



DIPARTIMENTO DI  
ELETTROTECNICA  
ED ELETTRONICA

# **POLITECNICO DI BARI**

**Titolazione:**

**INGEGNERIA ELETTRICA**

**Tesi:**

**CONTROLLO DI POSIZIONE E VELOCITÀ DI UN  
MOTORE A CORRENTE CONTINUA CON SCHEDA A  
MICROPROCESSORE dSPACE**

**Oihan Uriz Larrea**

**Prof. Francesco Cupertino**

**Bari, Luglio 2009**

*A Francesco Cupertino, per il suo aiuto e pazienza*



## Indice

### **Capitolo 1**

1.1	Introduzione.....	5
1.2	L'azionamento elettrico.....	6

### **Capitolo 2**

2.1	Introduzione al motore DC.....	8
2.2	Il motore a spazzole DC.....	11
2.3	Il modello matematico.....	15
2.4	Il sistema di controllo.....	18

### **Capitolo 3**

3.1	Introduzione al controllore PID.....	20
3.2	Caratterizzazioni del controllore PID.....	22
3.3	Metodi di taratura dei PID.....	25

### **Capitolo 4**

4.1	L'Unità d'Space.....	28
4.2	Finalità della tesi.....	37

### **Capitolo 5**

5.1	Il controllo di velocità.....	40
5.2	Il controllo de posizione.....	57
5.3	Conclusioni e sviluppi futuri.....	66

<b>Appendice A:</b>	The d'SPACE DS1104 R&D Controller Board (inglese).....	67
---------------------	--	----

<b>Appendice B:</b>	Control Design for Servo Amira DR300 (inglese).....	75
---------------------	---	----

<b>Bibliografia</b> .....	81
---------------------------	----

## CAPITOLO 1

### 1.1 Introduzione

In svariate attività industriali, in modo particolare nel settore della robotica, si presenta la necessità di controllare la traiettoria degli attuatori. I principali obiettivi che si cercano di raggiungere nel progetto di un controllo del movimento e quindi della traiettoria sono sostanzialmente due:

- Controllo della velocità durante il movimento del carico
- Esatto posizionamento del carico

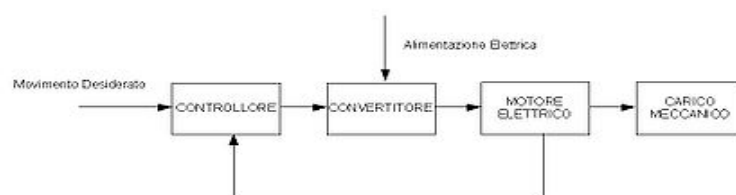
Tali obiettivi sono molto importanti in un ciclo produttivo qualsiasi, giacché il raggiungimento degli stessi comporta un risparmio economico e una riduzione dei tempi morti durante le fasi del ciclo, quindi una maggiore produttività a parità di tempo, a evidente vantaggio dell'industria che si serve di tali macchine. Si pensi ad esempio al lavoro eseguito da una macchina utensile che effettua e a quanto sia importante il controllo di velocità e di posizione.



Questo insieme di problematiche rende quindi, necessario il ricorso all'automatizzazione delle operazioni effettuate per ottimizzare le prestazioni e conseguentemente, lo sviluppo di un'opportuna metodologia di controllo. Lo scopo di questa tesi è fornire una tecnica di controllo adeguata per la soluzione dei problemi su citati, e in particolare per l'inseguimento di traiettoria di un motore in corrente continua utilizzando un triplo anello di retroazione al fine di controllare la posizione.

## 1.2 L'azionamento elettrico

Gli azionamenti elettrici sono sistemi che mediante un attuatore (motore) trasformano (in modo reversibile) l'energia elettrica in meccanica, avvalendosi di un'apparecchiatura elettronica di potenza (convertitore) atta a modulare l'energia elettrica, in accordo a una funzione di comando (controllo della coppia, della velocità o della posizione) che risponde a esigenze elaborate localmente (là dove si trova l'azionamento) oppure a un programma elaborato da un'intelligenza remota (chiamata a coordinare più azionamenti o processi di produzione). Il sistema di controllo rappresenta l'intelligenza locale che, basandosi sulle misure elettromeccaniche e i comandi che giungono attraverso l'interfaccia con il mondo esterno, è in grado di elaborare i comandi per modulare l'energia elettrica. Il convertitore elettronico riceve i comandi dal sistema di controllo ed eroga l'energia elettrica al motore elettrico controllando tensione e/o corrente per ottenere la coppia, la velocità, o la posizione desiderata. L'attuatore converte l'energia elettrica, modulata in energia meccanica, e agisce sul sistema tramite la trasmissione meccanica. Lo schema di principio di un generico azionamento elettrico è rappresentato *in figura 1.2.1*



**Figura 1.2.1**

Per l'ottenimento di un profilo di moto assegnato occorre variare e controllare la posizione, la velocità e l'accelerazione, legate dalle leggi della dinamica del moto; sono interessate anche le coppie motrici e resistenti e le inerzie dei componenti. Considerando che una traiettoria di moto descrive le modalità con cui l'organo controllato deve spostarsi da una posizione a un'altra, è evidente che l'anello di controllo di posizione angolare  $\varphi$  è il più esterno. L'anello è governato dal riferimento di posizione (set point) e impiega

un opportuno trasduttore di posizione (encoder incrementale). L'errore  $\varepsilon_p$ , elaborato e amplificato dall'amplificatore regolatore, comanda gli stadi successivi. In teoria, se l'obiettivo dell'azionamento è il *controllo della posizione angolare*  $\varphi$  dell'albero del motore, l'anello di controllo di posizione dovrebbe essere sufficiente per comandare l'intero azionamento. Tuttavia, internamente all'anello di posizione, si trova *di norma* anche la catena di controllo della velocità angolare. Questo secondo anello di controllo, il cui segnale di riferimento è prodotto dall'amplificatore regolatore di posizione, ha la funzione di impedire che, in presenza di elevati valori dell'errore di posizione  $\varepsilon_p$ , la velocità angolare possa crescere oltre la velocità nominale del motore. Sarà dunque sufficiente limitare opportunamente il valore del riferimento di velocità. Per raggiungere il nostro obiettivo, si è utilizzato un controllore PI (effettua un'azione proporzionale e un'integrale) per ogni anello di retroazione, una dinamo tachimetrica, un trasduttore di posizione, un motore DC, un *experiment laboratory* DR300 Amira e un Unità DSPACE per un controllo in *real time* del sistema.



## CAPITOLO 2

### 2.1 Introduzione al motore DC

Uno dei primi motori elettromagnetici rotanti, se non il primo, è stato inventato da Michael Faraday nel 1821, e consisteva in un filo conduttore tenuto fermo verticalmente alla sua estremità superiore in modo che l'estremità inferiore fosse immersa in un piatto contenente mercurio. Un magnete permanente circolare era sistemato al centro del piatto. Quando una corrente elettrica era fatta scorrere nel filo, questo ruotava attorno al magnete mostrando che la corrente generava un campo magnetico attorno al filo.

Il **motore in corrente continua** (brevemente **motore in CC**) è utilizzato ampiamente per piccole e grandi potenze. Sono a corrente continua (o comunque alimentabili in corrente continua) numero si motori di piccola potenza per usi domestici, come anche motori per trazione ferroviaria e marina della potenza di molte centinaia di KW.

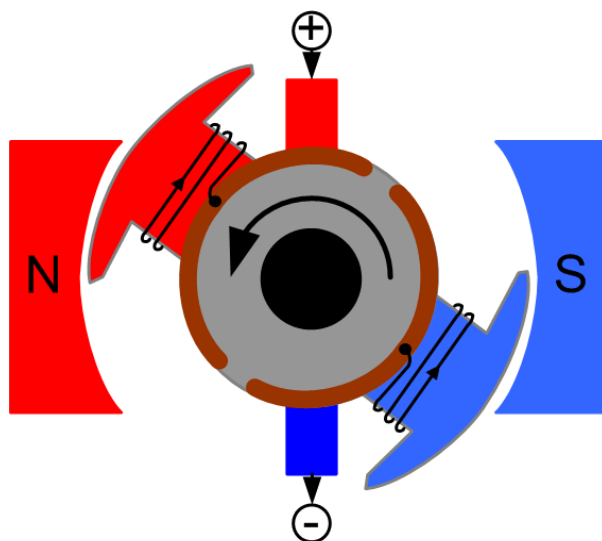
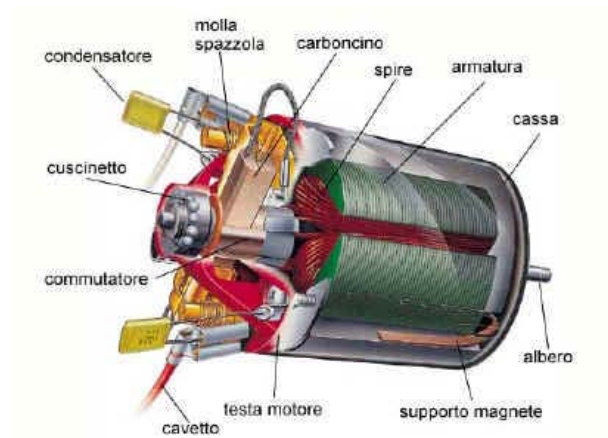


Figura 1.2.1 Un semplice motore DC.

Il motore elettrico, in genere, si compone di due parti: un *rotore* e uno *statore*: Il rotore è la parte mobile del motore, e comprende tutti quei conduttori destinati a stabilire un campo magnetico che dovrà interagire con un campo analogo prodotto dallo statore; il rotore, inoltre, possiede dei dispositivi che permettono di



connetterlo elettricamente con lo statore. Lo statore, *a sua volta*, contiene conduttori destinati a produrre un campo magnetico che dovrà interagire con quello prodotto dal rotore; in più lo statore si trova saldamente collegato a una base che sostiene l'intera apparecchiatura .



**Figura 2.2 Tipico motore a spazzole**

Il funzionamento di un motore elettrico in corrente continua si basa su due leggi fondamentali.

La prima legge, che deriva dalla legge di Faraday-Lenz, afferma che:

“Se si fa muovere un conduttore di sezione costante e di lunghezza  $l$ , con una velocità  $v$ , in un campo magnetico, si genera una forza elettromotrice indotta nel conduttore pari a:

- $E = B_n \cdot l \cdot v$

dove  $B_n$  è la componente di  $B$  perpendicolare a  $v$ . Il verso di  $E$  è calcolato usando la regola della mano destra.

La seconda legge afferma che:

" Se un conduttore a sezione costante di lunghezza  $l$ , immerso in un campo magnetico è attraversato da una corrente  $I$ , genera una forza:

- $F = B_n \cdot l \cdot I$

dove  $B_n$  è la componente di  $B$  perpendicolare ad  $I$ . La direzione e il verso di  $F$  si ottengono con la regola della mano sinistra.

Per meglio comprendere l'applicazione di queste due leggi sul motore elettrico bisogna sapere che, sia sullo statore che sul rotore, sono montati degli avvolgimenti di materiale conduttore, opportunamente alimentati.

Alcuni avvolgimenti hanno il compito di generare il campo magnetico  $B$ , formando il circuito di eccitazione; altri creano la sede di corrente  $I$ , formando il circuito di armatura. In generale, i due tipi di avvolgimenti si possono trovare sia sullo statore che sul rotore. Quando la corrente scorre negli avvolgimenti, si genera un campo magnetico intorno al rotore. La parte sinistra del rotore è respinta dal magnete di sinistra ed attratta da quello di destra. In maniera analoga, il magnete di destra, respinto dalla parte destra del rotore, viene attratto da quella di sinistra. Con tale interazione, la coppia di forze in gioco genera la rotazione.

## 2.2 Il motore DC a spazzole

Nei motori DC a spazzole, un interruttore rotante, detto *commutatore* o *collettore a spazzole* fissato all'albero rotante, con due contatti striscianti + e - (spazzole) collegati alla parte ferma, inverte il senso della corrente che percorre gli avvolgimenti generando un campo magnetico in verso o nell'altro secondo la direzione della corrente. Da queste continue inversioni nascono forze di attrazione e di repulsione con i magneti permanenti fissi.

Le virtù del motore DC a spazzole sono:

- hanno un elevato rapporto potenza/peso.
- sono relativamente facili da regolare, sia come velocità sia come coppia, soprattutto se non sono richieste prestazioni elevate.
- hanno un'elevata coppia di spunto che li rende ideali, per esempio, nella trazione elettrica.

Sono propriamente del motore a corrente continua le seguenti mancanze:

- le spazzole sono oggetti particolarmente fragili ed inaffidabili nel lungo periodo.
- la commutazione meccanica delle spazzole causa picchi di tensione elevata che mettono a dura prova l'elettronica ed aumentano le emissioni elettromagnetiche.
- le spazzole limitano la massima velocità a poche migliaia di giri al minuto.
- la coppia d'inerzia è relativamente elevata .

Le tensioni e le correnti che possono essere applicate ad un motore in corrente continua devono rimanere entro specifici limiti, che rappresentano i loro valori nominali o di targa, oltre i quali gli avvolgimenti del motore sarebbero eccessivamente sollecitati per un corretto e prolungato funzionamento. Ad esempio la corrente di armatura  $i_a$  dovrà avere ampiezze in valore assoluto non superiori al valore nominale  $i_{an}$  oltre il quale le perdite Joule che si producono nel circuito indotto porterebbero la temperatura di regime di questo componente ad assumere valori inaccettabili per i materiali isolanti. Solo per brevi intervalli si ammettono correnti maggiori della nominale (*limite di corrente nel funzionamento intermittente*).

Anche la tensione alle spazzole deve rimanere entro il suo valore nominale  $V_{an}$ , che dipende dai criteri d'isolamento adottati.

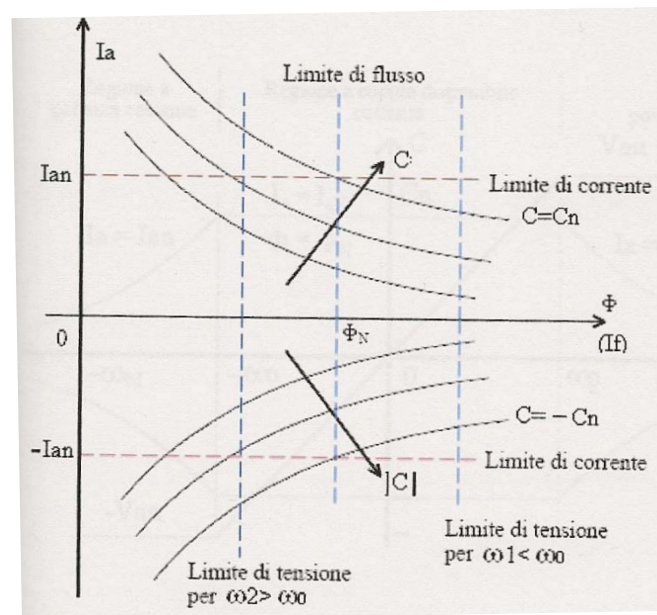
Infine anche per il circuito di campo saranno definite la corrente nominale  $i_{fn}$  e la corrispondente tensione nominale  $V_{fn}$ . Con tali valori nominali si produce il flusso nominale  $\Phi_n$  per il quale è stato dimensionato il circuito magnetico del motore. In sintesi i limiti di funzionamento a regime si potranno esprimere con:

$$|I_a| \leq I_{aN}$$

$$|V_a| \leq V_{aN}$$

$$|\Phi| \leq \Phi_N$$

I limiti sopra esposti producono a loro volta limiti di coppia e di velocità. Tutto questo può essere studiato con l'ausilio di un diagramma  $\Phi$ - $I_a$ , come in **figura 2.2.1**, sul quale sono tracciati *limite di corrente di armatura* e *il limite di flusso induttore*.

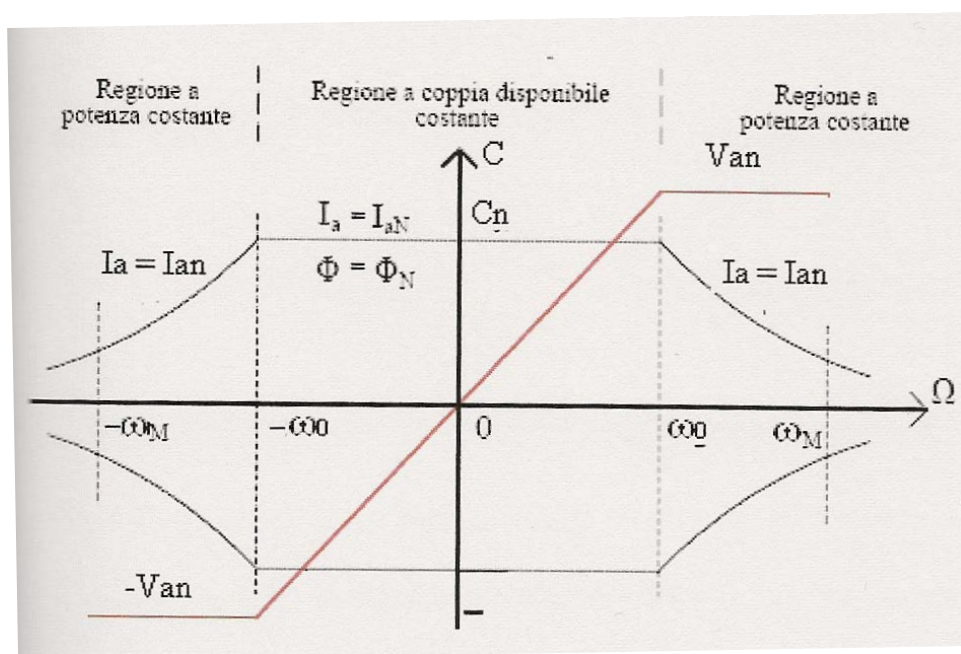


**Fig. 2.2.1**

Il limite di tensione può essere espresso come:

$$|Ra I_a + K\Phi\omega| \leq V_{aN} \quad \Phi \leq \text{---} \quad (2.2.1)$$

Si può notare che il limite di tensione dipende dalla velocità del motore. Per velocità inferiori alla velocità a vuoto il limite di tensione eccede quello di flusso. Ciò vuol dire che dovendo rispettare il limite di flusso, la tensione non raggiunge il suo valore nominale. Per velocità invece superiori alla velocità a vuoto il limite di tensione eccedono quello di flusso per cui quest'ultimo dovrà essere ridotto. Sul diagramma  $I_a - \Phi$  sono state tracciate anche le *curve a coppia costante* (iperbole) per individuare quale flusso convenga produrre nel motore. Il motore, dato una certa corrente di armatura, produce la massima coppia disponibile quando il flusso è massimo. La conseguenza di ciò è che per velocità inferiori alla velocità a vuoto il motore lavorerà a flusso costante e nominale, rendendo così disponibile la coppia nominale, con corrente di armatura nominale. La regione di funzionamento con velocità inferiori alla velocità a vuoto prende il nome di regione a coppia disponibile costante o a flusso costante. Il tutto può essere osservato in **figura 2.2.2** :



**Figura 2.2.2**

Per velocità superiori alla velocità a vuoto il motore lavorerà con il più alto valore di flusso ammesso inversamente proporzionale alla velocità e tale da produrre sempre una tensione ai morsetti, costante e pari al suo valore nominale. La coppia disponibile, sempre con la corrente nominale è in questo caso decrescente con la velocità come il flusso. Trascurando le perdite, la potenza meccanica disponibile, data dal prodotto

$C\omega$  è costante. La regione di funzionamento con velocità superiori alla velocità a vuoto prende per questo il nome di regione a *potenza disponibile costante o deflussaggio*.

Come si può vedere, le prestazioni del motore elettrico sono limitate dalla presenza delle spazzole. La soluzione a tale problema sta nell'uso di motori senza spazzole (*brushless*), i quali complicano irrimediabilmente l'elettronica della macchina .

## 2.3 Il modello matematico del motore DC

Per la macchina a corrente continua abbiamo le seguenti equazioni.

Circuito d'armatura:  $V_a = R_a * i_a + L_a \frac{di_a}{dt} + e$  (2.3.1)

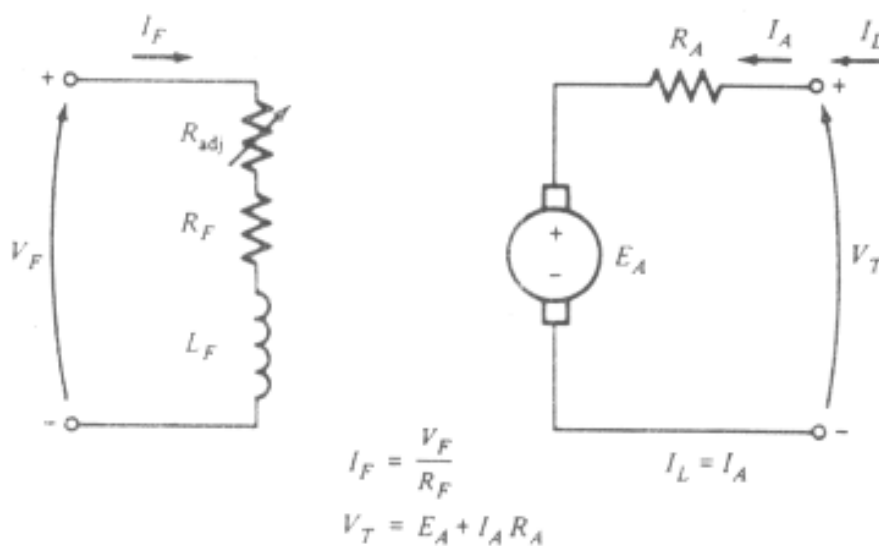
Circuito di eccitazione:  $V_f = R_f * i_f + L_f \frac{di_f}{dt}$  (2.3.2)

Forza contro elettromotrice:  $e = K \omega$  (2.3.3)

Coppia elettromotrice:  $C = K \omega * i_a$  (2.3.4)

Equilibrio meccanico:  $J \frac{d\omega}{dt} = C - C_r$  (2.3.5)

Raddrizzatore trifase:  $V_a = V_r * K_a$  (2.3.6)



**Figura 2.3.1** Circuito elettrico motore DC

In situazione di regime , le derivate sono nulle —

Quindi da 2.3.1:

$$V_a = R_a * i_a + e$$

$$i_a = \text{---} \quad (2.3.7)$$

$$(2.3.4) \quad C = K \quad i_a \quad = (\text{da } 2.3.7) \quad K \quad \text{---} = (\text{da } 2.3.3) \quad K \quad \text{---} = \text{---} - \omega$$

$$= (\text{da } 2.3.6) \quad \text{---} - \omega$$

$$\omega = \text{---} - \text{---} * C \quad (2.3.8)$$

Da questa equazione facciamo lo schema a blocchi:

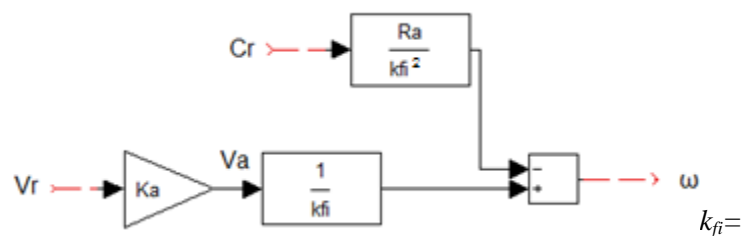


Figura 2.3.2

Con Cr nulla  $\omega = (\text{da } 2.3.8) \text{---} \longrightarrow V_r = \text{---} \omega^* \quad \omega = \omega^* = \omega \text{ desiderata}$

Con Cr non nulla  $\omega < \omega^*$

L'obiettivo da raggiungere è uguagliare  $\omega$  a  $\omega^*$



Per lo studio in condizioni transitorie dobbiamo applicare la trasformata di Laplace alle equazioni (2.3.1),(2.3.2),(2.3.3),(2.3.4),(2.3.5). Si ottiene:

$$\omega(s) = \text{—————} \quad (2.3.9)$$

$$C(s) = K \Phi * i_a(s) \quad (2.3.10)$$

$$i_a(s) = \text{—————} \quad (2.3.11)$$

$$e(s) = K \omega(s) \quad (2.3.12)$$

dove  $t_a = L_a/R_a$  , costante di tempo elettrica dell'avvolgimento d'armatura.

Quindi:

$$(2.3.9) \quad \omega(s) = \text{—————} = (\text{da 2.3.10}) \quad \omega(s) = \text{—————} = (\text{da 2.3.11}) \quad \omega(s) = \text{—————}$$

$$= (\text{da 2.3.12}) \text{ ed } (\text{da 2.3.8.})$$

$$\omega(s) = \text{—————}$$

## 2.4 Il sistema di controllo

Lo schema a blocchi per l'anteriore equazione sarà

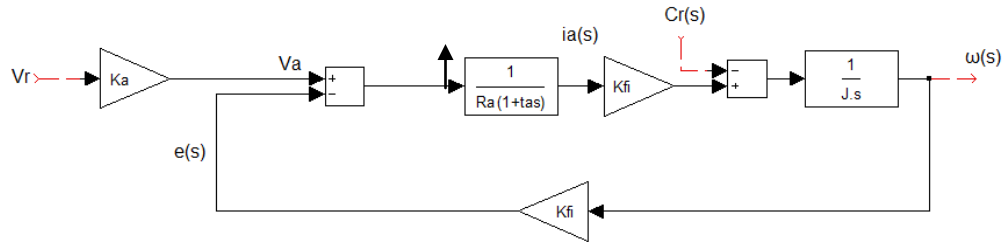


Figura 2.4.1

Si cerca ora di intervenire sul sistema per modificarne ulteriormente la dinamica. S'individua, inizialmente, la funzione di trasferimento in cui l'uscita è rappresentata da  $i_a$ , e non come in precedenza da  $\omega$ .

Si considera prima l'ingresso  $V_a$  con  $C_r$  nulla:

$$G(s) = \frac{1}{Ra(1+tas)}; H(s) = Kfi; (s) = \frac{1}{J.s} = \frac{1}{J.s}$$

$$i_a(s)_1 = \frac{1}{Ra(1+tas)} V_a(s);$$

e poi l'ingresso  $-C_r$  con  $V_a$  nulla:

$$G(s) = -Kfi; H(s) = \frac{1}{J.s}; (s) = \frac{1}{J.s} = \frac{1}{J.s}$$

$$i_a(s)_2 = -Kfi C_r(s);$$

Quindi, l'uscita totale sarà:

$$i_a(s) = i_a(s)_1 + i_a(s)_2 = \frac{1}{Ra(1+tas)} V_a(s) - \frac{Kfi}{J.s} C_r(s)$$

Ridisegniamo lo schema mettendo in evidenza anche la corrente  $i_a$  oltre che  $\omega$

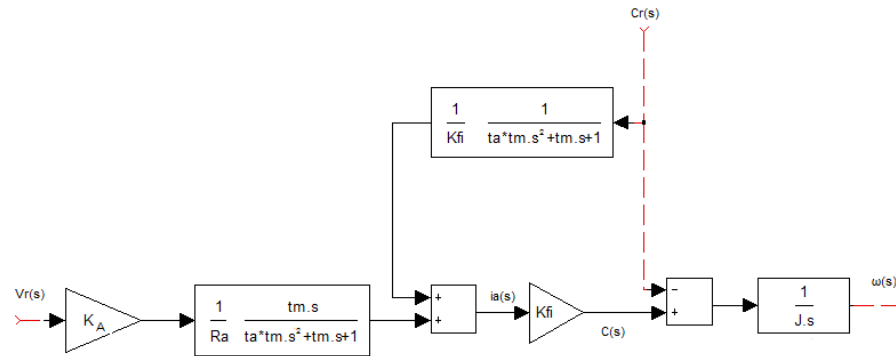


Figura 2.4.2

L'obiettivo da raggiungere è il controllo della dinamica della corrente e la velocità. Procediamo retroazionando il segnale  $i_a$  (anello di corrente) e il segnale  $\omega$  (anello di velocità). Inseriamo, per il controllo degli anelli, i regolatori PI: Per il segnale  $i_a$ , è necessario il filtro per diminuire il disturbo, e per il segnale  $\omega$  usiamo il filtro per ridurre il ripple di tensione all'uscita della dinamo.

Per rappresentare più correttamente il raddrizzatore trifase utilizziamo la funzione di trasferimento

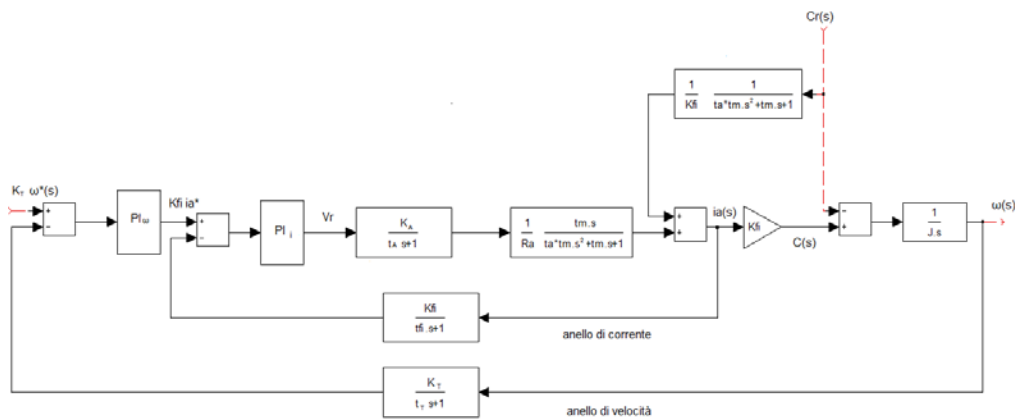


Figura 2.4.3

### 3.1 Introduzione al controllore PID

- Prima di tutto il loro impiego consente di controllare in modo soddisfacente un'ampia gamma di processi;
- In secondo luogo, negli anni sono state sviluppate e largamente utilizzate semplici regole per la loro taratura automatica, applicabili con buoni risultati anche nel caso in cui non sia disponibile un modello matematico preciso del sistema sotto controllo;
- Inoltre, per la loro semplicità, i *PID* possono essere realizzati con le tecnologie più varie: Meccaniche, pneumatiche, idrauliche, elettroniche analogiche e digitali.

$$\text{---} \quad \text{---} \tag{3.1.1}$$

20

Nel dominio di Laplace, l'espressione del regolatore PID analogico ideale diventa:

$$U(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) E_d \quad (3.1.2)$$

### 3.2 Caratterizzazioni del controllore PID

La funzione di trasferimento  $U(s)/E(s)$  del regolatore in (3.1.2) la indicheremo con  $G_{PID}(s)$  :

$$G_{PID}(s) = K_p \left( 1 + \frac{1}{T_I s} + T_D s \right) = K_p + \frac{K_p}{T_I s} + K_p T_D s = K_p \left( 1 + \frac{1}{T_I s} + T_D s \right) \quad (3.2.1)$$

$K_p$ ,  $K_I$  e  $K_D$  prendono il nome, rispettivamente, di coefficiente dell'azione proporzionale, coefficiente dell'azione integrale e coefficiente dell'azione derivativa ;

tempo integrale:  $T_I = K_p/K_I$

tempo derivativo:  $T_D = K_D/K_p$

I tipi di regolatori che possiamo avere sono:

Regolatore Proporzionale (P):

$$G_P(s) = K_p \quad (3.2.2)$$

Regolatore Integrale (I):

$$G_I(s) = \frac{K_p}{s} \quad (3.2.3)$$

Regolatore Proporzionale Integrale (PI):

$$G_{PI}(s) = K_p \left( 1 + \frac{1}{T_I s} \right) \quad (3.2.4)$$

Regolatore Proporzionale Derivativo (PD):

$$G_{PD}(s) = K_p (1 + T_D s) \quad (3.2.5)$$

Regolatore Proporzionale Integrativo Derivativo (PID):

$$G_{PID}(s) = K_p \left( 1 + \frac{1}{T_I s} + T_D s \right) \quad (3.2.6)$$

### L 'azione Proporzionale (P)

- Per avere un errore a regime basso è necessario occorre un guadagno d'anello elevato, occorre che sia elevato il coefficiente dell'azione proporzionale ( $K_P$ )
- Possibili problemi riguardanti la stabilità del sistema a catena chiusa
- Vantaggio dovuto al fatto che un valore elevato del coefficiente dell'azione proporzionale riduce gli effetti delle variazioni parametriche e dei disturbi
- Dal punto di vista dinamico l'azione proporzionale produce un aumento della banda passante, quindi una migliore prontezza di risposta, ma allo stesso tempo riduce i margini di stabilità.

### L 'azione Integrale (I)

- La funzione principale dell'azione integrale è quella di annullare l'errore a regime in presenza di disturbi costanti sulla variabile d'uscita.
- Dal punto di vista dinamico l'azione integrale porta ad un peggioramento dei margini di stabilità in quanto introduce nella funzione di trasferimento a catena aperta un ritardo di fase pari a  $\pi/2$ .
- Riduzione della banda passante.

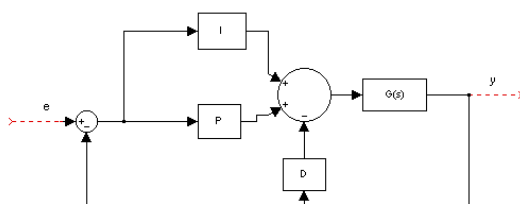
### L 'azione Derivativa (D):

- Migliora i margini di stabilità in quanto introduce un anticipo di fase pari a  $\pi/2$ .
- Introduce un' azione di controllo proporzionale alla variazione dell' errore, fornendo, quindi, una correzione che anticipa l'andamento dell'errore nel tempo.
- Svantaggio legato all'aumento della banda passante che porta ad amplificare i segnali con contenuto armonico a frequenze elevate come ad esempio il rumore sovrapposto al segnale utile .
- Non è fisicamente realizzabile in quanto la risposta in frequenza ad essa associata ha modulo crescente al crescere della frequenza
- Nella pratica i regolatori PID sono caratterizzati da un'azione derivativa definita dalla seguente funzione di trasferimento:

$$D(s) = \frac{N}{s^N}$$

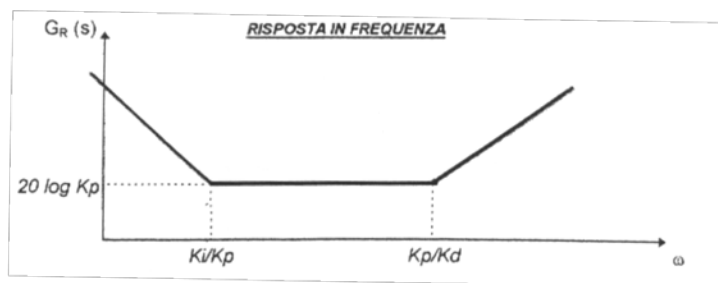
dove  $N$  è un parametro il cui valore viene posto pari a  $10 \div 100$  in modo tale che il polo di valore  $N/T_D$ , introdotto per ottenere la fisica realizzabilità, si collochi al di fuori della banda di frequenze di interesse per il progetto del sistema di controllo.

- sovente l'azione derivativa viene imposta soltanto sulla variabile d'uscita  $y$  (fig. 3.2.1)



**Figura 3.2.1**

Come si può osservare nella funzione di trasferimento (3.2.1) infatti il numeratore di  $G_{PID}(s)$  è di grado superiore al denominatore pertanto, così come scritta, la funzione di trasferimento non è fisicamente realizzabile. Ciò corrisponde all'impossibilità di ottenere dall'errore un segnale che ne costituisca in ogni istante la derivata. Per rendere realizzabile l'azione derivativa occorrerà in effetti aggiungere un polo in alta frequenza, per altro di norma irrilevante ai fini della valutazione delle prestazioni del regolatore PID. Dall'ultima espressione scritta per  $R(s)$  si riconosce che al variare di  $T_I$  e  $T_D$  gli zeri del regolatore possono essere reali o complessi e coniugati. Imponendo la presenza di due zeri reali e distinti, naturalmente nel semipiano sinistro, il diagramma di Bode del modulo di  $G_{PID}$  assumerà l'andamento tipico riportato in Figura(3.2.2):



**Figura 3.2.2**



### 3.3 Metodi di taratura dei PID

Il progetto del regolatore PID si riduce quindi alla scelta del guadagno e della posizione degli zeri. Per quanto riguarda il controllo in cascata, di velocità e posizione, del motore a corrente continua, oggetto di questa tesi, si sono utilizzati, come descritto nel capitolo 2, dei regolatori PI. Questi sono alcuni metodi per progettare i regolatori:

#### -Metodo del modulo ottimo e metodo dell'ottimo simmetrico.

Questo metodo è il più utilizzato per la taratura dei PID, e sarà anche utilizzato per progettare i nostri regolatori per il controllo della velocità.

Il metodo del modulo ottimo garantisce la riduzione della sovraelongazione al 4,3% del valore di regime, ma non ha una buona reiezione al disturbo (coppia non nulla).

Il metodo dell'ottimo simmetrico, consente di conseguire una buona reiezione al disturbo, ma applicato un set point di velocità a gradino, la sua sovraelongazione sarà molto accentuata (circa il 43%).

Per ovviare a tale inconveniente, già progettato il regolatore con il metodo dell'ottimo simmetrico. Si inserirà un filtro di smoothing a valle del set point per ridurre notevolmente la sovraelongazione senza modificare la reiezione al disturbo.

L'unico problema da risolvere è la eccessiva corrente assorbita all'avviamento. Questo si supererà implementando un limitatore di corrente.

Questo metodo può vedersi più dettagliato nel esperimento che sarà realizzato.

#### -Metodo in frequenza con Diagramma di Bode

Un altro metodo molto utilizzato nella taratura dei controllori è la taratura in frequenza attraverso il diagramma di Bode. Si realizza osservando e modificando il valore del margine di fase nel diagramma di Bode per la scelta dei guadagni dei regolatori.

#### Metodo di Ziegler e Nichols in frequenza

Il metodo di Ziegler-Nichols, risalente al 1942, è tra i più usati ed è apprezzato per la sua semplicità, per il fatto di non richiedere un modello matematico del processo, e per le prestazioni che riesce a produrre.

Il sistema reazionato è stabile ad anello chiuso solo se il guadagno in continua ad anello aperto è mantenuto ad un certo livello (che dipenderà dal sistema). Aumentando il guadagno, nei sistemi a stabilità il sistema diventa instabile. Tra la zona di stabilità (poli della funzione di trasferimento ad anello chiuso con componente reale negativa) e la zona di instabilità (poli con componente reale positiva) esiste un limite (poli con componente reale nulla) in cui il sistema non diverge, ma oscilla con ampiezza di oscillazione costante. Quindi, se aumentando il guadagno in continua si arriva ad una situazione di oscillazione stabile, possiamo assumere che un valore di guadagno nettamente inferiore a tale limite possa corrispondere ad una condizione di lavoro ragionevole. La procedura da seguire per l'ottimizzazione del controllore è la seguente:

- Annullare, o quantomeno ridurre al minimo livello possibile, le azioni di integrazione e derivata facendo lavorare il controllore in modo puramente proporzionale.
- Operando ad anello chiuso, aumentare progressivamente (e prudentemente) il valore di  $K_p$  fino a che si osserva che l'uscita  $y(t)$  oscilla in modo permanente (limite di stabilità). Indichiamo con  $K_O$  il valore limite di  $K_p$ . Indichiamo con  $T_O$  il periodo di oscillazione misurato quando  $K_p = K_O$ . Se il sistema non entra mai in oscillazione, il metodo non è applicabile.
- A seconda della configurazione del controllore, applicare i parametri riportati nella tabella sottostante.

	$K_p$	$T_i$	$T_d$
P	$0.5K_{cr}$	$\infty$	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

**Figura 3.3.14**

Per concludere, ricordiamo alcuni limiti pratici dei metodi di tuning dei controllori appena descritti. Noi abbiamo sempre assunto che i processi da controllare siano lineari e stazionari. Purtroppo tali approssimazioni non sono sempre valide. In particolare, uno degli errori più comuni è quello di ottimizzare i parametri del controllore quando il processo da controllare si trova in condizioni molto diverse da quelle operative. In tal caso eventuali non-linearità possono giocare un ruolo importante e l'ottimizzazione iniziale può perdere di significato. In altri casi le oscillazioni che devono essere innescate per tarare il controllore secondo il metodo di Ziegler e Nichols in frequenza possono provocare danni al processo stesso. Quindi, nella pratica, bisogna operare con prudenza ricordando che i metodi semi-empirici appena descritti forniscono una ottimizzazione solo parziale del controllore. Il risultato finale dipende strettamente dal comportamento dinamico del processo sotto controllo.

## CAPITOLO 4

### 4.1 L'Unità DSPACE

*dSpace* è il marchio di una società tedesca, operante nel settore delle soluzioni di sviluppo rapido per sistemi di controllo di apparati meccanici ed elettronici, che ha prodotto pacchetti *software* e schede di prototipazione per microprocessori ad elevata capacità di elaborazione. L'unità di sviluppo in questione, consiste in un pacchetto comprendente la scheda *DS1104*, che ospita due processori, diversi dispositivi logici programmabili e circuiti per la gestione dei segnali da e verso periferiche. Comprende, inoltre, un *software* per la programmazione della scheda per il monitoraggio delle grandezze in ingresso o in uscita, derivanti dal pannello di interfaccia con segnali analogici e digitali. Tale pacchetto ha lo scopo principale di consentire la prototipazione rapida di algoritmi di controllo su piattaforma; *TI DSP TMS320F240* tale unità in virgola mobile è considerata, ad oggi, proponibile economicamente anche in applicazioni di media complessità, Esso è un mezzo molto usato nel campo industriale, soprattutto per eseguire simulazioni in tempo reale sui macchinari. La versione utilizzata non è delle più recenti e neanche delle più evolute in termini di velocità e di gestione dei canali I/O. È richiesto un PC con processore 603e / 250MHz, o superiore su cui è installato l'applicativo *Matlab*. In ambiente di simulazione *SIMULINK* si programma molto semplicemente la scheda, grazie ai *Tools* messi a disposizione, e si possono eseguire test simulativi *off-line* e test in *real-time*.

#### LA SCHEDA DS1104

L'*hardware* fornito dalla *dSpace* è essenzialmente costituito dalla scheda *DS1104*, progettata per sviluppare controlli digitali di elevate prestazioni e simulazioni in tempo reale in vari campi tecnologici, quali:

- Robotica;
- Attuatori elettrici e servo idraulici;
- Controlli per pilotaggio unità dischi;
- Controlli per autoveicoli;
- Controlli per sistemi vibranti;
- Calcoli su segnali digitali.

La struttura della scheda è molto compatta ed è munita di connessioni per essere inserita negli slot della scheda madre di un PC. Essa monta a bordo il *DSP TMS320F240* (Digital Signal Processor) il quale rappresenta il nucleo più importante dell'unità.

Il *DSP* è stato potenziato da una circuiteria che gestisce l'interfacciamento con alcune periferiche esterne, quali *encoders* per il rilevamento di posizione o velocità, come unità di elaborazione o dispositivi analogici, e che gestisce l'interfacciamento con i cinque canali d'ingresso/uscita analogici, e i 20-bit di ingresso/uscita digitali. I cinque canali analogici d'ingresso comprendono l'unità di conversione analogica/digitale (ADC) a 16-bit con tempi di conversione di 2 $\mu$ s e 4 unità di conversione analogico/digitale a 12-bit con tempi di conversione di 800 $\mu$ s (la velocità della conversione è ovviamente inversamente proporzionale alla risoluzione del dato). Il range di tensione massima ammissibile è di  $\pm 10V$  con un *offset* introdotto di  $\pm 5mV$ .

Le tensioni massime in uscita sono ancora di  $\pm 10V$ . I canali di interfacciamento comprendono anche due ingressi provenienti da altrettanti *encoders* incrementali, ognuno dei quali fornisce ben sei segnali, tra i quali un indice di riferimento per il contatore a 24-bit, presente sulla porta stessa, per l'inizio del conteggio. La massima velocità di conteggio è di 6.6MHz.

I segnali sono anche filtrati, per eliminare i disturbi indotti sul cavo che collega il sensore ottico al pannello dei connettori della scheda *DSPACE*. L'acquisizione viene effettuata con un protocollo RS422 compatibile. Esso fornisce anche l'alimentazione 5V e 0.5A per l'*encoder*.

La scheda è inoltre munita di un sottosistema I/O digitale programmabile, gestito completamente da un altro processore che è un *DSP* a virgola fissa.

Le alimentazioni richieste dalla scheda sono:

- 5V  $\pm$  5%, 2,5A
- $\pm 12V \pm$  5%, 0.3A.

## IL PANNELLO DEI CONNETTORI I/O

Il pannello dei connettori ingresso/uscita è necessario per collegare facilmente la scheda *DS1104* alle periferiche che inviano i segnali all'azionamento o li prelevano dai sistemi di misura. In questo modo possiamo connettere qualsiasi strumento o dispositivo evitando di eseguire saldature.

## INTERFACCIA MATLAB/DSPACE

Come detto in precedenza, il *software* messo a disposizione della *Dspace* consente di interfacciare il *DSP* con il programma *Matlab* ed in particolare con il suo strumento di programmazione grafica e simulazione *SIMULINK*.

Attraverso la libreria *RTI* (Real Time Interface) di oggetti *I/O Interface* si interfaccia il modello con i canali di ingresso/uscita del dispositivo stesso. L'aspetto più rilevante è che si riescono ad effettuare simulazioni in *Real-Time* sul sistema avviando semplicemente la simulazione del modello editato in ambiente grafico *SIMULINK*.

Il segnale in ingresso ad ogni canale è compreso tra  $\pm 10V$ , corrispondente a  $\pm 1$  nel blocco *SIMULINK*. Per questo motivo è necessario moltiplicare per dieci i dati in uscita dai quattro canali del blocco. Se non tutti i canali sono necessari, è consigliabile terminare le uscite non utilizzate.

In aggiunta ai blocchi di Simulink, usati per realizzare la simulazione dell'azionamento, ci sono dei blocchi di interfaccia che collegano la scheda della *dSPACE* con il mondo esterno. Questi blocchi sono contenuti nel software di *Matlab dSPACE RTI 1104*. Di seguito se ne riporta una breve descrizione:

- Il blocco ***DS1104ADC\_Cx***, è utilizzato per l'acquisizione delle correnti e della tensione  $V_{dc}$ . Esso consente di leggere da uno solo dei quattro canali convertitori A/D del pannello connettore, inoltre effettua la conversione tra segnale in ingresso e segnale in uscita secondo la tabella 4.1.



Input Voltage Range	Simulink Output
-10..... +10	-1..... +1

**Tab 4.1**

- Il blocco **DS1104DAC\_Cx**, permette di scrivere in uno solo degli otto canali convertitori D/A del anello connettore. Esso converte il segnale d'ingresso in una tensione analogica in uscita secondo la tabella 4.2



Simulink Input	Output Voltage Range
-1..... +1	-10..... +10

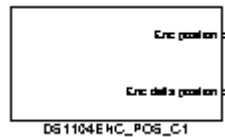
**Tab 4.2**

Nel caso specifico, il blocco rende disponibile in uscita il riferimento di coppia per il carico, quando questo è controllato in remoto.

- Il blocco **DS1104ENC\_SETUP**, setta i parametri globali dei due canali dell'encoder. Esso deve essere situato nel modello per poter utilizzare tutti gli altri blocchi di interfaccia dell'encoder. Per i due canali possono essere scelti due tipi di segnali:
  - Differential (RS422);
  - Single-ended (TTL).



- Il blocco **DS1104ENC\_POS\_Cx**, legge le posizione e la variazione della posizione di uno dei due canali dell'encoder. Permette inoltre di inizializzare la posizione dell'encoder all'inizio della simulazione



## CONTROLDESK

La caratteristica più interessante della scheda in esame è che il pacchetto software di questa (il ControlDesk) ha la capacità di interagire con il pacchetto software di calcolo numerico Matlab. Tale iterazione permette di programmare la scheda con l'intuitivo sottopacchetto di *Matlab*, *Simulink*, (pacchetto per simulazioni con interfaccia utente a blocchi che lavora in associazione con *Matlab* mediante delle specifiche librerie) mentre altrimenti ci si sarebbe dovuto accontentare di realizzare complicati listati in *C* mediante il compilatore nel *ControlDesk* o altri compilatori.

La scheda, per ragioni storiche, è stata realizzata per un'iterazione ottima con ogni versione di *Matlab*.

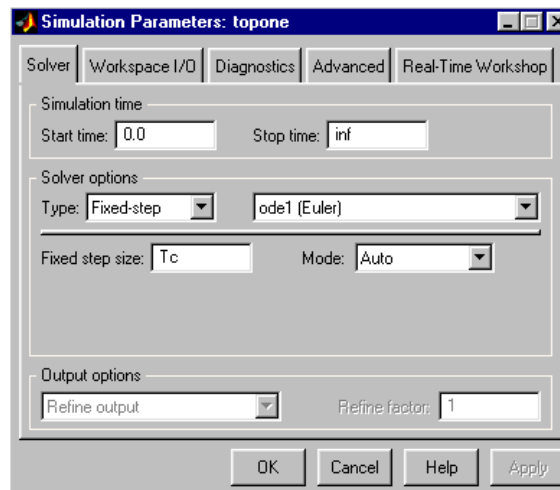
Il termine programmare, nel titolo del paragrafo, non è usato a caso: alla fine di questa esposizione si sarà realizzato, oltre a un diagramma a blocchi di *Simulink*, un listato in *C* che può essere comodamente letto nel *ControlDesk* e un file .object che dovrà essere inserito nella memoria della scheda *DSP* affinché faccia quanto realizzato in *Matlab*. Aprendo un nuovo file *Simulink* si può, mediante la pratica libreria <nomelibreria> con i blocchi che costituiscono le parti della *DSP*, realizzare un diagramma a blocchi rappresentativo di ciò che si vuole ottenere mediante la scheda. È bene tenere presente quali blocchi sono di input e quali di output per una buona lettura dei diagrammi realizzati, ma per questo basta guardare i data sheet. Quando il diagramma è stato realizzato, bisogna effettuare la fase di compilazione, generando i file che ci permetteranno di portare lo schema di controllo realizzato sul *ControlDesk*. I file hanno lo stesso nome che è stato dato al file contenente il diagramma a blocchi di *Simulink*, ma con estensione .*SDF* e .*PPC*.



Per compilare lo schema basta effettuare le seguenti operazioni:

1. Dal menù bar di *Simulink* selezionare *Tools* e quindi *RTW Options* che apre una nuova finestra per impostare i parametri della simulazione e, quindi, della compilazione
2. Selezionando *Solver* tramite *Type*, selezionare *Fixed-step*, mentre mediante *Fixed step size* selezionare un adeguato passo di campionamento.
3. Fatto ciò, selezionare *Apply* e chiudere la finestra.
4. Settati i parametri di *real-time*, selezionando nuovamente *Tools* e cliccare su *RTW Build*.

Verranno così compilati i suddetti files.

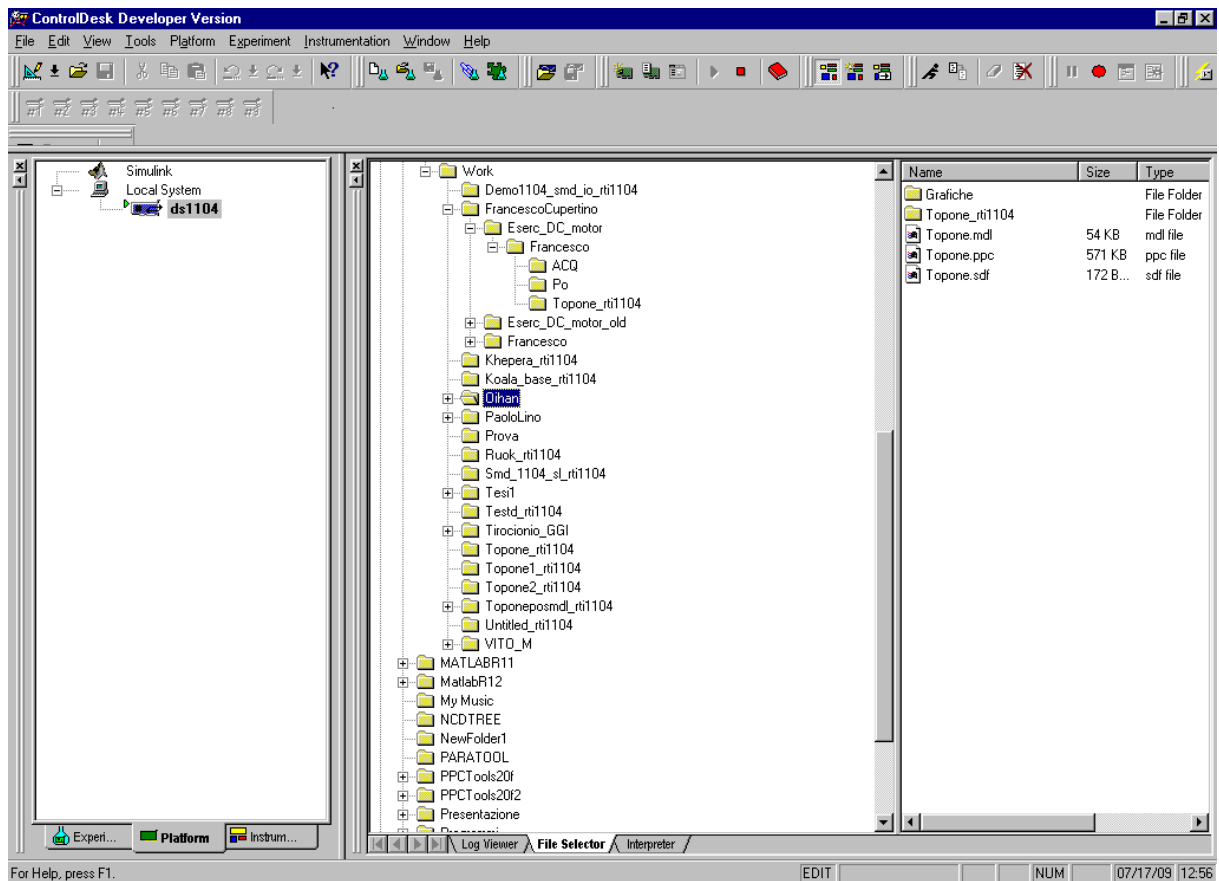


Dopo aver realizzato i files con *Matlab*, per trasferire l'esperimento al *ControlDesk* bisogna aprirlo per mandare in memoria alla scheda *DSP* il programma compilato e per creare pratiche interfacce grafiche.

Partendo dal *Navigator* iniziale, basta andare sulla sezione *Hardware* ed eseguire i seguenti punti:

1. Visto che il file era in *Matlab*, i files della compilazione saranno nella cartella *Work* di *Matlab*
2. Aprire tale cartella dal browser messo a disposizione dal *Navigator* e trascinare sull'icona *DS1104* il file *.obj* che è stato realizzato in precedenza
3. Selezionando *View* e successivamente *ToolWindow*, appare una nuova finestra.

All'interno di essa selezionare la sezione *Model*, e in questo modo, sulla destra appariranno le variabili che sono state utilizzate da *Simulink*. Tali variabili sono configurabili e regolabili mediante il *ControlDesk*, rendendo anche la gestione dell'esperimento delegabile a questo programma di interfaccia. Si mostra, allora, come è possibile tale gestione .



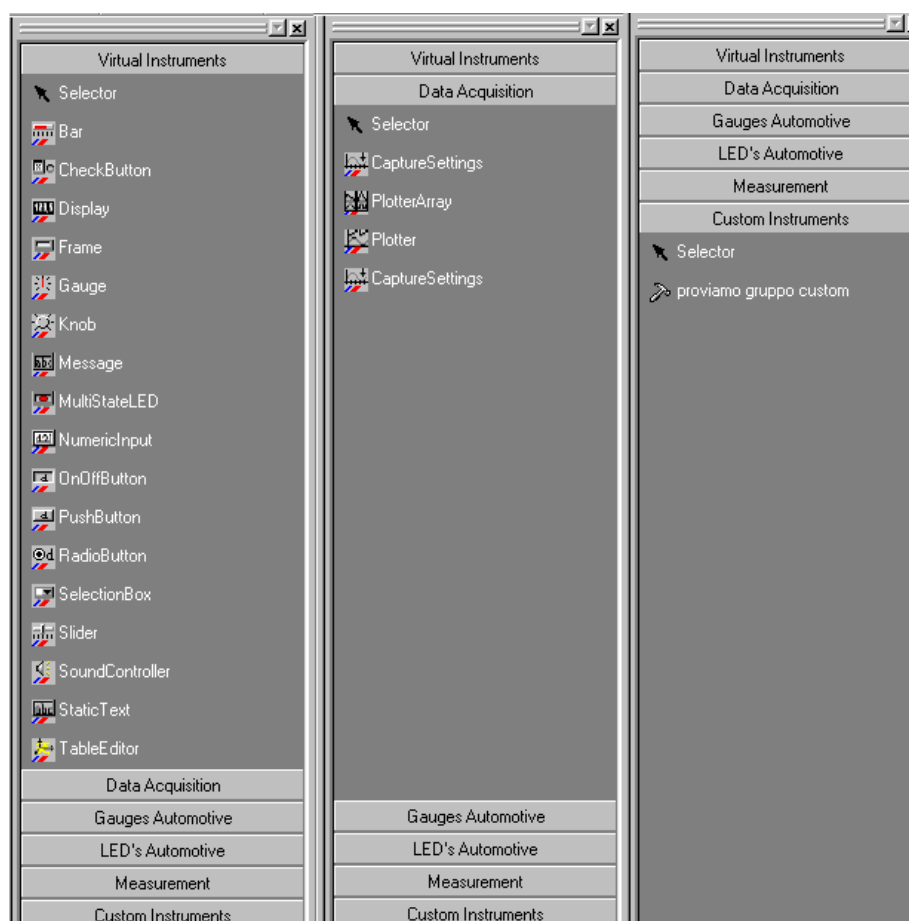
Per gestire quanto fatto mediante *ControlDesk*, bisogna creare un nuovo esperimento. Ciò è possibile selezionando, dal menù bar del programma, *File*, quindi *New Experiment*. Dopo di che va creato un nuovo foglio di lavoro dove andranno posizionati gli elementi grafici che possono essere utilizzati per la gestione del *DSP*. Tale foglio può essere creato dai passi:

1. Selezionare *File* dal menù bar del *ControlDesk*, poi *New*.
2. Uscirà una finestra di dialogo che permette di scegliere tra varie voci. Selezionare *Layout*.

Questo apre due nuove finestre, una denominata *Layout* e l'altra *Instrument Selector*.

Per creare uno strumento all'interno del foglio bianco bisogna:

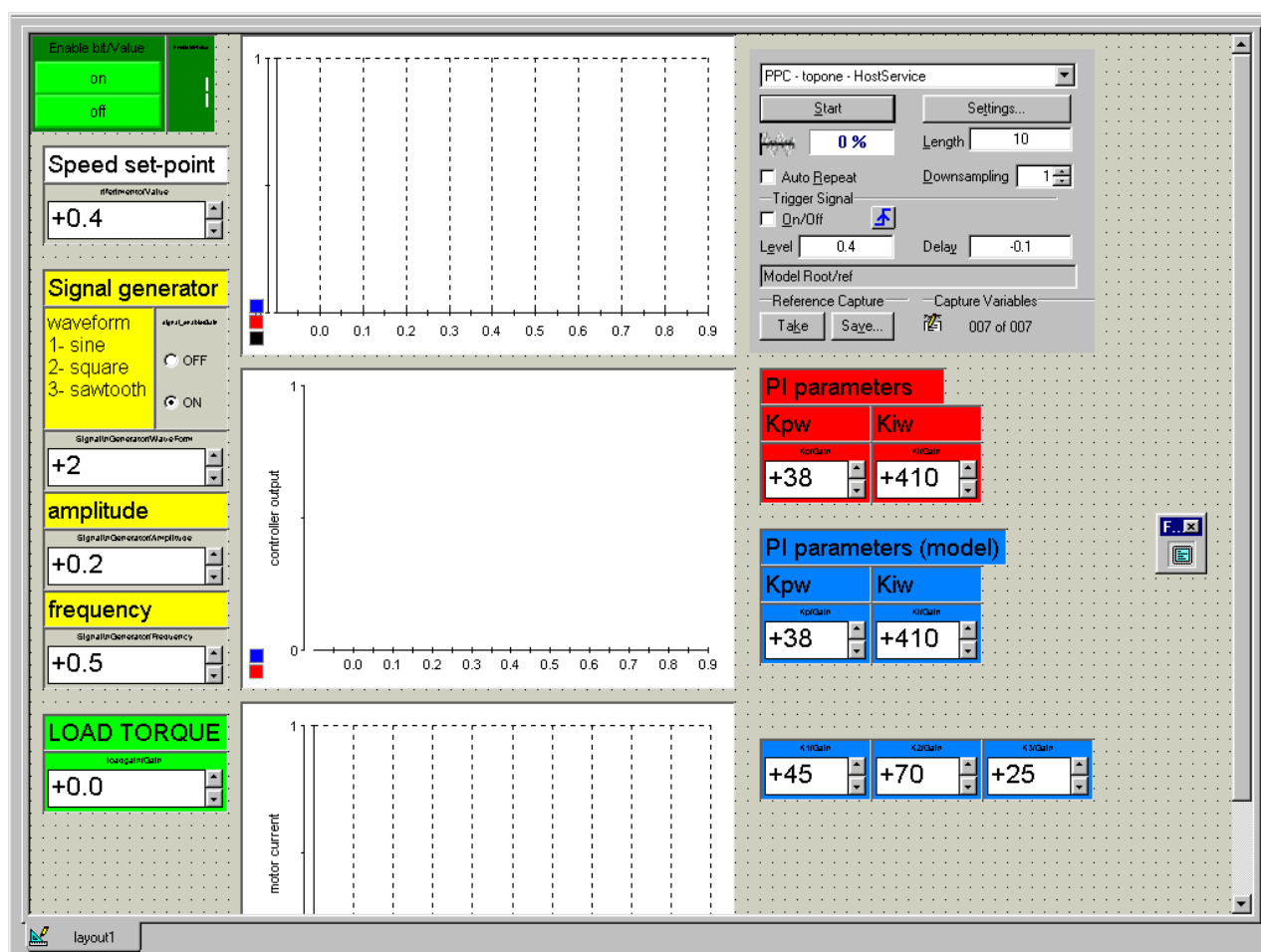
1. Cliccare sul gruppo di strumenti che contiene lo strumento che si desidera inserire (ad esempio *Data Acquisition*)
2. All'interno di *Instrument Selector* fare doppio click sull'icona dello strumento che si vuole realizzare (ad esempio *Plotter*) e nella finestra *Layout* disegnare un rettangolo della dimensione desiderata per lo strumento, usando il mouse.
3. Appena creato, lo strumento avrà un contorno rosso che sta ad indicare che non è stato associato nessuna variabile. dell'esperimento. Per fare ciò bisogna prendere una variabile dal Variable Browser e trascinarla sopra lo strumento con il mouse.



Una volta inseriti tutti gli strumenti desiderati selezionare dal menù bar *File* e quindi *Save*, per salvare il nuovo *Layout*. Tornando al *Navigator*, selezionare la sezione *Experiment*, andare sul nome dell'esperimento in oggetto. Cliccando col tasto destro del mouse, selezionare *Add All Opened Files* per aggiungere il *Layout*

all'esperimento. In questo modo, ogni volta che si riavvierà l'esperimento, saranno già disponibili gli strumenti creati.

Per avviare l'esperimento basta andare su Instrumentation del menú. bar del ControlDesk e successivamente su Animation mode. In questo modo il DSP comincerà a gestire i dati in real-time e ad eseguire il programma.



## 4.2 Finalità della tesi

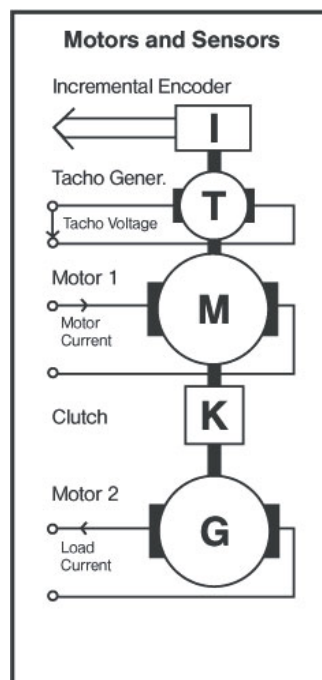
La finalità di questa tesi è stata quella di definire e progettare, una soluzione per il controllo di posizione ed inseguimento di traiettoria, di un motore a corrente continua, attraverso un controllo in cascata composto da tre anelli di retroazione.

Partendo da un sistema di controllo di velocità in cascata, come descritto nei capitoli precedenti, è stato aggiunto un terzo anello di retroazione, per il controllo della posizione.

Le prove sperimentali sono state effettuate nel laboratorio di robotica, con l'ausilio di un laboratorio sperimentale AMIRA DR300 composto come in figura.



[ close ]

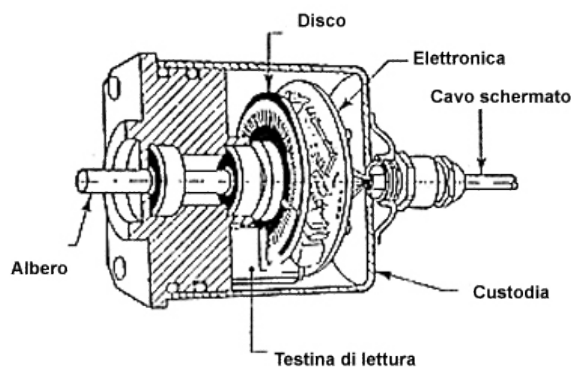


dove il Motor1 rappresenta il nostro motore a corrente continua a spazzole ,vero è proprio, il Motor2 è anch'esso un motore elettrico ma in questo contesto viene utilizzato per simulare una coppia di carico .Con T si rappresenta la dinamo tachimetrica , un dispositivo a che si collega meccanicamente all'albero del motore e che fornisce in uscita una tensione continua proporzionale alla velocità di rotazione della dinamo stessa. La possiamo vedere in figura 4.2.1

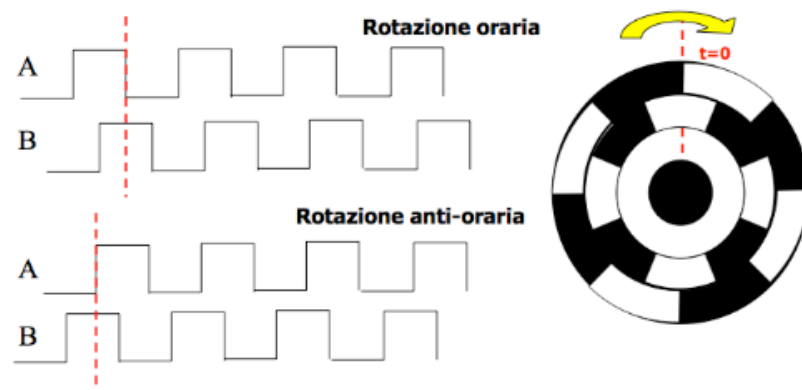


**Figura 4.2.1**

Con I si rappresenta l'encoder incrementale. Esso è un dispositivo digitale che, collegato direttamente all'albero del motore, permette di misurarne con estrema precisione la posizione angolare. Nel nostro caso l' encoder ha una sola uscita che, per ciascun giro dell'albero emette 1024 impulsi digitali. Possiamo vedere il funzionamento in figura 4.2.2 4.2.3:



**Figura 4.2.2**

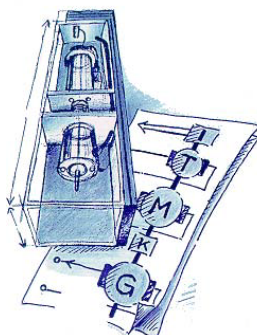


**Figura 4.2.3**

Con K infine si rappresenta il dispositivo di frizione.

Gli anelli di controllo della corrente del motore e del carico sono integrati nel dispositivo di alimentazione. E' possibile fornire dall'esterno (nel nostro caso utilizzando la scheda DS1104) i riferimenti di corrente. Attraverso prove sperimentali è stata ricavata una funzione di trasferimento del secondo ordine che ben approssima il comportamento dinamico degli anelli di controllo della corrente. Tale funzione di trasferimento sarà utilizzata per la taratura degli anelli di controllo di velocità e posizione, implementati con la scheda della dSPACE.

Il diagramma a blocchi che corrisponde la sistema di controllo di posizione e velocità è stato realizzato in Simulink e poi caricato sulla scheda dSPACE 1104. Per verificare in tempo reale le prestazioni del sistema di controllo è stato utilizzato il software ControlDesk. Sono state realizzate delle interfacce grafiche per visualizzare in tempo reale la dinamica del sistema, variandone di volta in volta i parametri, quali ad esempio i guadagni dei regolatori PI per approssimare quanto più possibile la funzione di trasferimento del sistema di controllo reale al sistema di controllo realizzato.



## CAPITOLO 5

### 5.1 Il controllo di velocità

#### Introduzione

Per il controllo di velocità abbiamo costruito il diagramma, realizzato in *Simulink*, che possiamo vedere in figura 5.1.1:

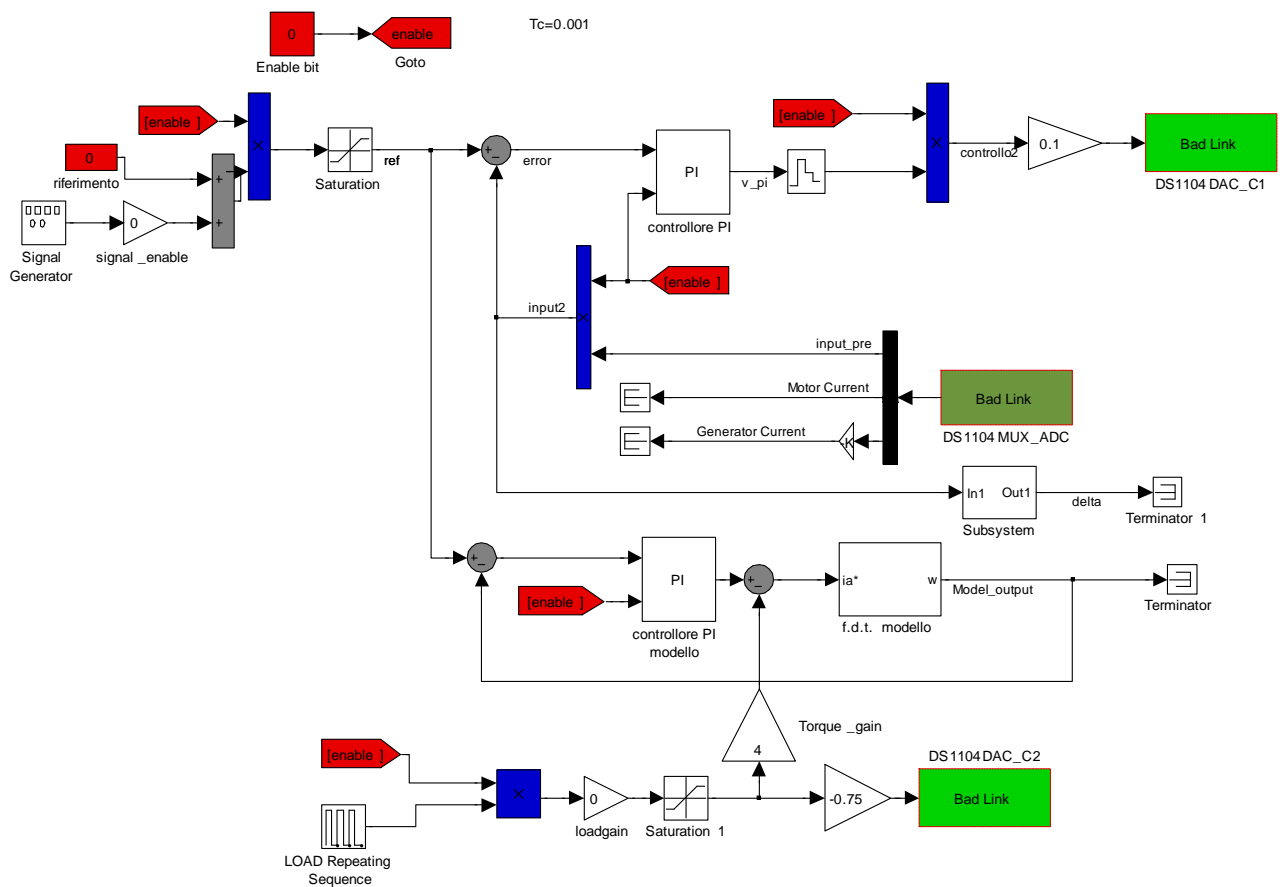


Figura 5.1.1

Possiamo vedere nella figura due parti fondamentali:

- Nella parte in alto è rappresentato il **diagramma reale del controllo del motore**.
- Nella parte di sotto vediamo il **diagramma modello del controllo del motore**.



Il diagramma reale è costituito dai due blocchi DS1104DAC\_C1 e DS1104MUX\_ADC1, che abbiamo visto nel capitolo precedente. Sono rispettivamente:

- Un convertitore di segnale digitale- analogico per trasmettere la segnale digitale alla *dSpace*, la quale comunica con il laboratorio sperimentale AMIRA DR300, e quindi con il motore.
- Un convertitore analogico- digitale per leggere i segnali provenienti dai sensori collocati sull'AMIRA DR300, come la velocità letta dalla dinamo, o la posizione del encoder.

Nel diagramma del modello è presente una funzione di trasferimento che deve rappresentare tutto che abbiamo nel diagramma reale fra i due blocchi spiegati prima.

I blocchi PI sono i regolatori che dobbiamo progettare (al inizio sono tarati con valori  $k_p = 50$  e  $k_i = 250$ ). Il blocco di saturazione serve per contenere la corrente al avviamento, e così proteggere la macchina da sovracorrenti troppo elevate, giacché la massima corrente ammissibile è 1.8 volte la corrente nominale. Tutto viene azionato con un generatore di segnale ad onda quadra, che ha un' ampiezza uguale a uno e frequenza 0.5 Hz, con un tempo di campionamento  $T_c = 0.001$ .

La interfaccia del *Control Desk* creata per controllare il esperimento è presentato in figura 5.1.2:

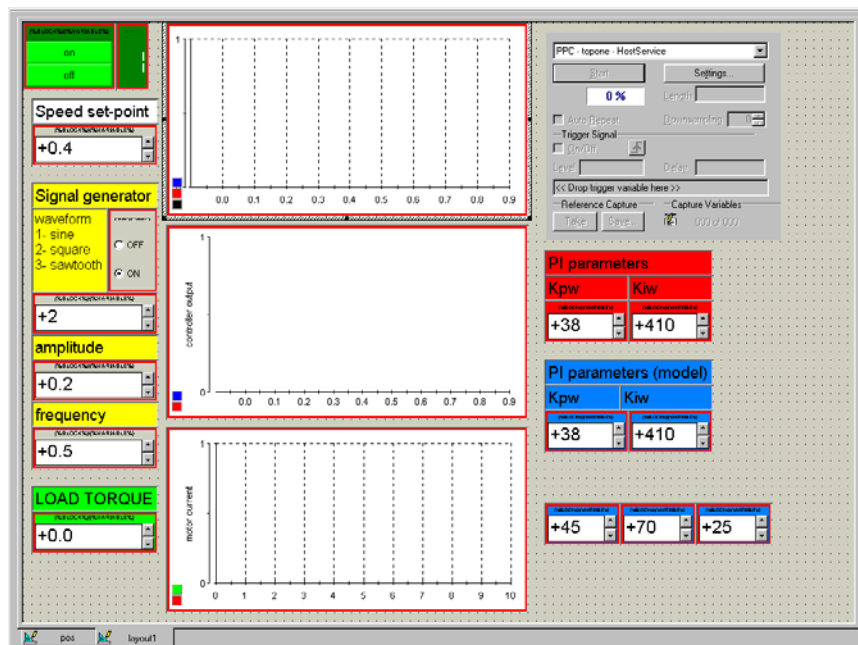


Figura 5.1.2

Qui si può modificare i parametri della funzione di trasferimento del diagramma del modello, i parametri dei controllori PI, il set point di velocità e la sua forma di onda e la coppia. Possiamo anche vedere e salvarle i dati per la realizzazione dei grafici.

### Descrizione del banco di prova

Il banco di prova è costituito da due macchine in corrente continua accoppiate meccanicamente. La prima macchina opera da motore ed è controllata in velocità, mentre la seconda funge da carico (generatore) ed è controllata in corrente. Le macchine in corrente continua utilizzate in questa esercitazione, oltre ad avere il consueto circuito di armatura, dalla cui corrente dipende la coppia sviluppata, hanno il circuito di eccitazione costituito da magneti permanenti, che generano il flusso necessario a eccitare la macchina. La presenza dei magneti permanenti fa sì che il flusso prodotto dal circuito di eccitazione sia costante.

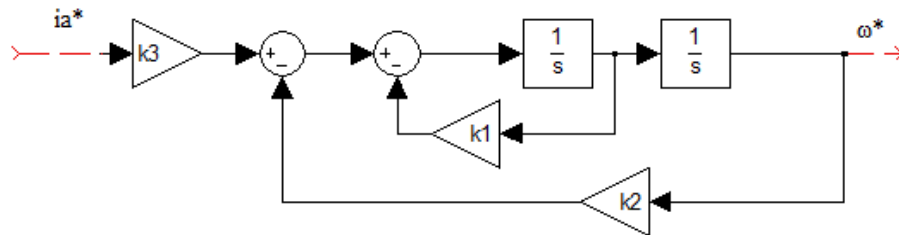
Sull'asse del sistema sono calettati una dinamo tachimetrica, che fornisce la misura della velocità e un encoder incrementale, che misura la posizione. Nell'esercitazione in questione il segnale dell'encoder incrementale non è di alcuna utilità, poichè il controllo di posizione non è implementato.

Per entrambe le macchine, l'anello di controllo della corrente è realizzato in maniera analogica, (tramite l'utilizzo di amplificatori operazionali) con schede installate in un box che riceve in ingresso i set point di corrente provenienti dalla scheda DSpace e fornisce in uscita la tensione con cui sono alimentati i circuiti di armatura delle due macchine. Tale box ha l'interruttore di accensione posizionato sul retro e per abilitarlo all'acquisizione dei segnali provenienti dalla scheda della DSpace si agisce sulla leva External.

Il controllo della velocità è stato realizzato in ambiente Matlab/Simulink ed è eseguito in tempo reale dalla scheda della DSpace.

### Modello del plant

Nel modello Simulink, che implementa il controllo di velocità di uno dei due motori in corrente continua, è stata inserita una funzione di trasferimento che modella il comportamento del sistema meccanico costituito dall'accoppiamento dei due motori.



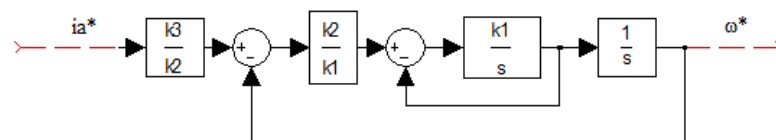
**Figura 5.1.3: Diagramma a blocchi del modello del plant.**

La presenza di tale modello consente di testare i parametri del regolatore, realizzando una simulazione numerica in parallelo all'esperimento. In tal modo è possibile valutare la risposta del sistema in simulazione e subito dopo sperimentalmente, valutando la coerenza tra i due test.

In figura 5.1.3 è riportato il diagramma a blocchi del modello del plant, mentre di seguito sono specificati i valori dei guadagni:

$$k_1 = 45 \quad k_2 = 70 \quad e \quad k_3 = 25$$

Per la progettazione del sistema di controllo è necessario mostrare la funzione di trasferimento del plant. Per questo si semplifica il diagramma a blocchi. Portiamo in catena diretta i blocchi presenti in retroazione. Con quest'operazione il diagramma a blocchi assume la struttura mostrata in figura 5.1.4.

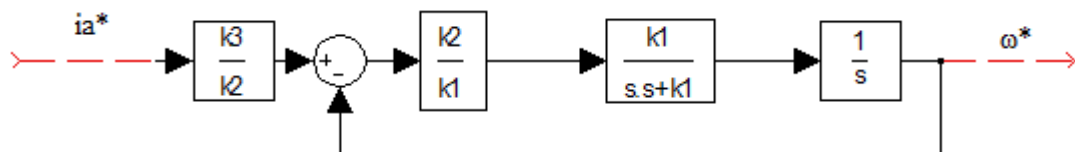


**Figura 5.1.4: Diagramma con i blocchi in catena diretta.**

In secondo luogo risolviamo gli anelli con retroazione unitaria, sapendo che la funzione di trasferimento è uguale a:

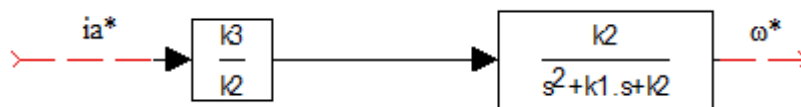
\_\_\_\_\_

dove  $G(s)$  è la funzione di trasferimento in catena diretta, mentre  $H(s)$  è la funzione di trasferimento sul ramo di retroazione. Risolviamo prima l'anello interno. Possiamo vedere la soluzione nella figura 5.1.5.



**Figura 5.1.5: Diagramma a blocchi dopo la risoluzione dell'anello interno.**

In figura 5.1.6 è mostrato il diagramma a blocchi dopo la risoluzione dell'anello esterno.



**Figura 5.1.6: Diagramma a blocchi dopo la risoluzione dell'anello esterno**

Semplificato già tutto lo schema, possiamo scrivere la funzione di trasferimento del plant.

$$G_{PLANT}(s) = \text{_____}$$

Sostituendo i valori numerici:

$$G_{PLANT}(s) = \text{_____}$$

Per una maggiore praticità di calcolo è preferibile esprimere il denominatore della funzione di trasferimento sottoforma di costanti di tempo. Per far ciò è necessario calcolare i poli della funzione di trasferimento.

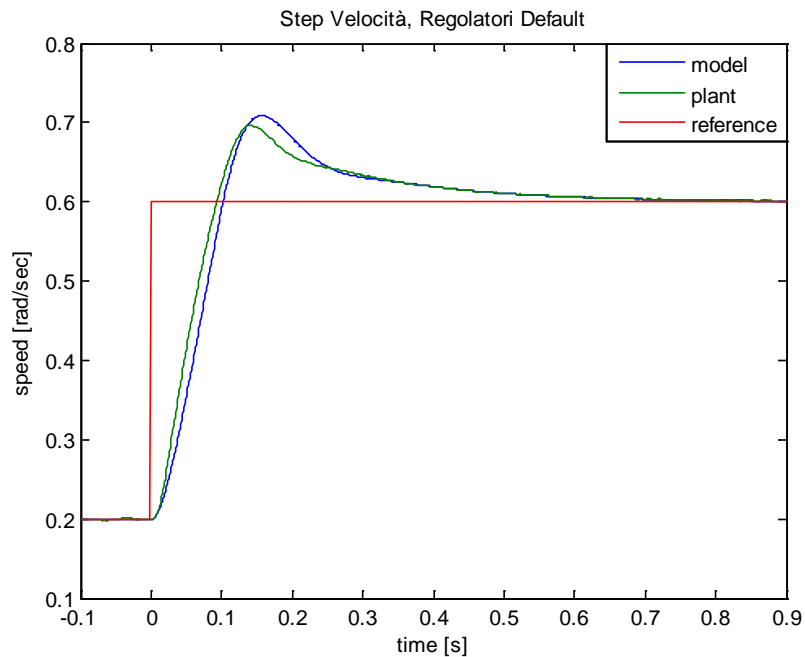
$$s^2 + 45s + 70 = 0 \quad \text{per} \quad s = -1,61 \quad e \quad s = -43,4$$

Le radici del denominatore sono reali e negative, per cui il sistema è stabile.

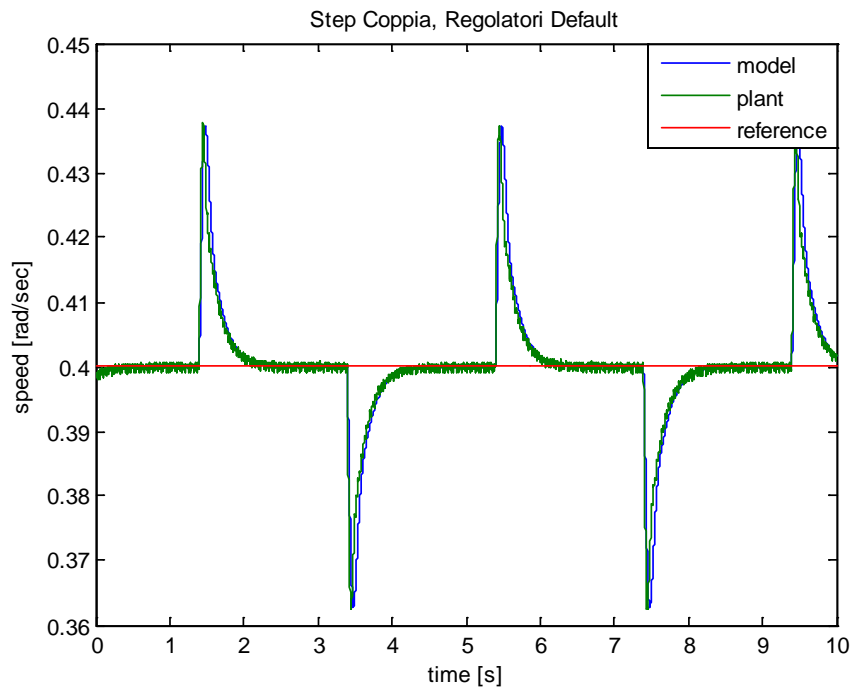
$$G_{PLANT}(s) = \frac{\quad}{\quad}$$

Il polo dominante è quello con costante di tempo 0,621.

Le figure 5.1.7 e 5.1.8 riportano rispettivamente la risposta al gradino di velocità e quella al gradino di coppia, quando i due regolatori hanno guadagni  $k_p = 50$  e  $k_i = 250$  (valori di partenza).



**Figura 5.1.7: Risposta a/gradino di velocità con  $K_p=50$  e  $K_i=250$ .**



**Figura 5.1.8: Risposta al gradino di coppia con  $K_p=50$  e  $K_i=250$ .**

### Criterio del modulo ottimo

Il criterio di progetto del modulo ottimo consente di contenere la sovraelongazione della risposta del sistema al gradino (4,3 % del valore di regime), ma non garantisce una buona reiezione al disturbo.

Nell'esercitazione in questione, in assenza di carico (disturbo), si applica come set point di velocità un gradino e si analizza la risposta del sistema reale e quella del sistema modellizzato mediante funzione di trasferimento.

Il regolatore di velocità, da progettare col criterio del modulo ottimo, è un regolatore PI, la cui funzione di trasferimento è:

$$G_{PI}(s) = k_p + \frac{k_i}{s} = k_p \left[ 1 + \frac{1}{Ts} \right] = k_p \left[ 1 + \frac{1}{Ts} \right] = k_p \frac{Ts + 1}{s}$$

dove  $Ts = k_p/k_i$  è la costante di tempo del regolatore.

La cascata in anello aperto tra il regolatore e il plant ha la seguente funzione di trasferimento:

$$G_{OL}(s) = k_p \frac{1}{s + \dots}$$

La prima fase di progetto, è accelerare la dinamica del sistema, praticando la cancellazione polo-zero per eliminare il polo dominante del plant. La cancellazione polo-zero viene effettuata attraverso l'imposizione della seguente uguaglianza:

per cui

Quindi è stata univocamente definita la costante di tempo del regolatore di velocità. Per effetto della cancellazione polo-zero la funzione di trasferimento in anello aperto, della cascata regolatore/plant diventa:

$$G_{OL}(s) = k_p \frac{1}{s + \dots} = \frac{1}{s + \dots}$$

La funzione di trasferimento del modulo ottimo in anello aperto è:

$$G_{mo}(s)_{OL} = \frac{1}{s + \dots}$$

Nota la funzione di trasferimento del modulo ottimo in anello aperto, s'impone l'uguaglianza tra i termini di pari grado.

$$\left\{ \begin{array}{l} 2\tau = \dots \end{array} \right.$$

Risolviamo il sistema di equazioni, sapendo che  $\tau = \dots$ , per effetto della cancellazione polo-zero, si ricava il valore del guadagno proporzionale.

$$k_p = \dots = 37,7$$

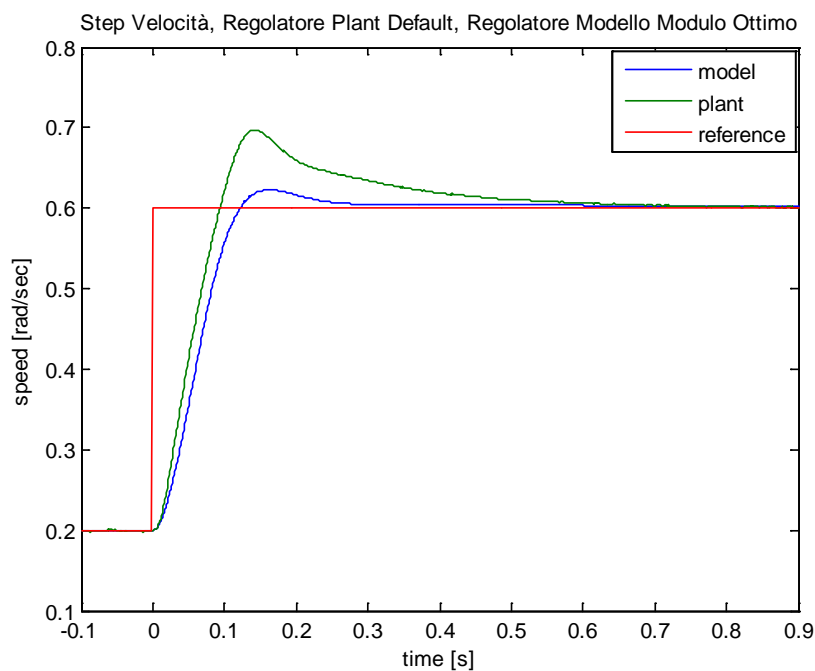
Per completare il progetto del regolatore è necessario determinare il valore del guadagno integrale:

$$k_i = \frac{1}{s} = 60,7$$

In conclusione, i guadagni del regolatore di velocità progettato, col criterio del modulo ottimo, sono:

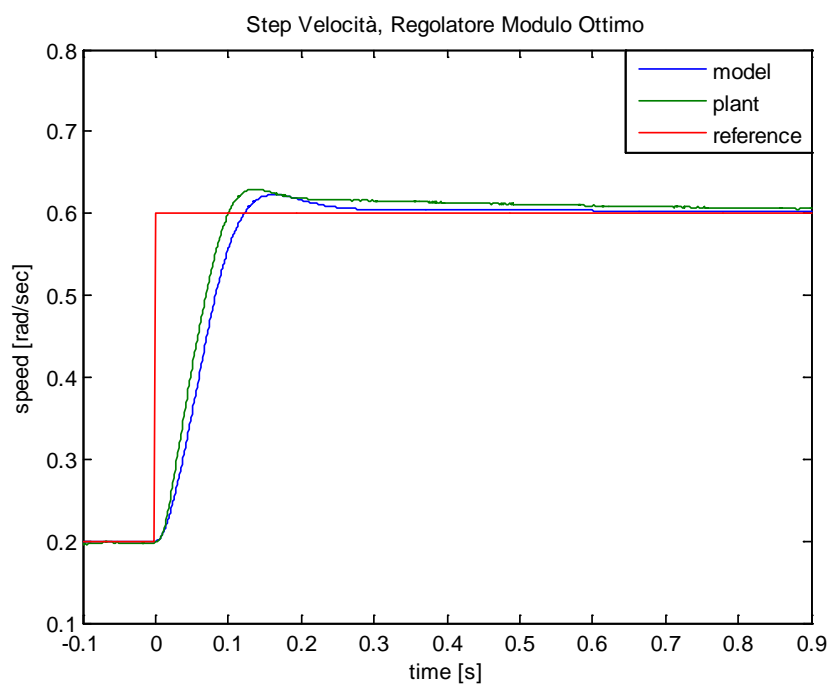
$$k_p = 37,7 \cong 38 \quad \text{e} \quad k_i = 60,7$$

Progettato il regolatore, col criterio del modulo ottimo, si testa prima sul modello (figura 5.1.9) e dopo sul motore reale (figura 5.1.10), in modalità controllo di velocità senza carico. Per verificare la reiezione al disturbo del modulo ottimo, si applica un gradino di coppia di carico (figura 5.1.11).

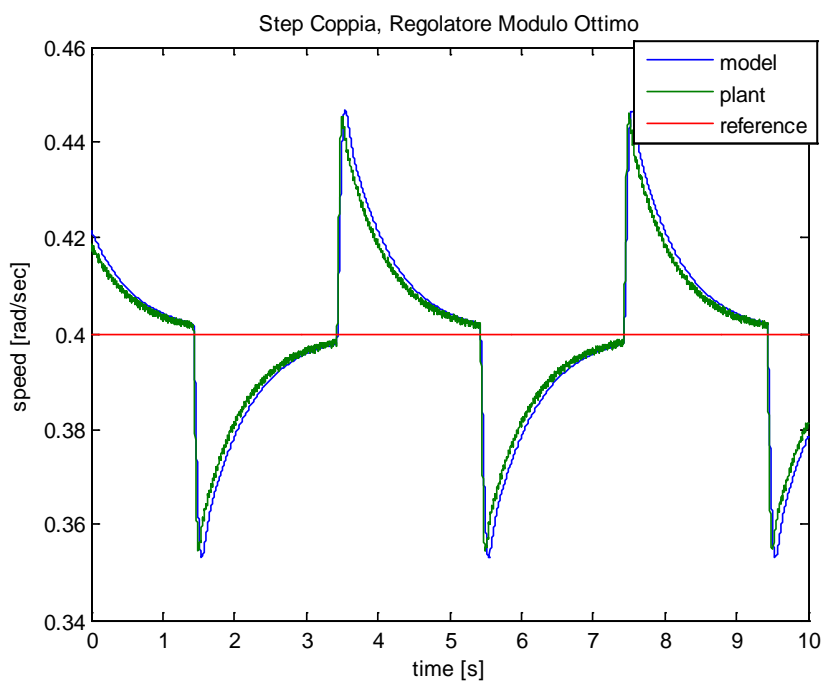


**Figura 5.1.9: Risposta al gradino di velocità con guadagni modulo ottimo (nel modello)**





**Figura 5.1.10: Risposta a/ gradino di velocità con guadagni modulo ottimo.**



**Figura 5.1.11: Risposta al gradino di coppia con guadagni modulo ottimo**

Possiamo vedere che il motore ha una buona risposta al gradino di velocità (la sovraelongazione non è troppo alta), ma non risponde bene al disturbo (con coppia non nulla).

### Criterio dell'ottimo simmetrico

L'ottimo simmetrico consente di conseguire una buona reiezione al disturbo. Per verificare la risposta del sistema al disturbo, si mantiene la prima macchina (motore) in controllo di velocità con set point costante e si applica alla seconda macchina (generatore) un set point di corrente ad onda quadra. La funzione di trasferimento (in anello aperto) dell'ottimo simmetrico è:

$$G_{OS}(s)_{OL} = \frac{1}{s^2 + 0,621s + 1}$$

Come già ricavato in precedenza, la funzione di trasferimento (in anello aperto) della cascata regolatore/plant è:

$$G_{OL}(s) = k_p \frac{1}{s^2 + 0,621s + 1}$$

Dall'analisi delle due funzioni di trasferimento si osserva come quella dell'ottimo simmetrico sia caratterizzata da due poli nell'origine, mentre l'altra ha solo un polo nell'origine. Per fare in modo che anche la funzione di trasferimento (in anello aperto) della cascata regolatore/plant abbia due poli nell'origine, si effettua la seguente ipotesi semplificativa:

$$0,621s + 1 \approx 1$$

ovvero si trascura l'unità rispetto alla quantità  $0,621s$ , che equivale ad approssimare il polo dominante ad un polo nell'origine.

Alla luce di tale assunzione, la funzione di trasferimento (in anello aperto) della cascata regolatore/plant diventa:

$$G_{OL}(s) = k_p \frac{1}{s} = \frac{1}{s}$$

Imponendo l'uguaglianza tra i termini di pari grado, si ottiene:

$$\begin{cases} 8 = \frac{1}{s} \\ 4 = \end{cases}$$

Si risolve sostituendo la prima equazione nella terza, ricavando così l'espressione della costante di tempo del regolatore di velocità:

$$\tau_{PI} = 4\tau = 4 \cdot 0.023 = 0.092$$

Effettuando le opportune sostituzioni, dalla seconda equazione si calcola il valore del guadagno proporzionale:

$$k_p = \frac{1}{0.023} = 37,7$$

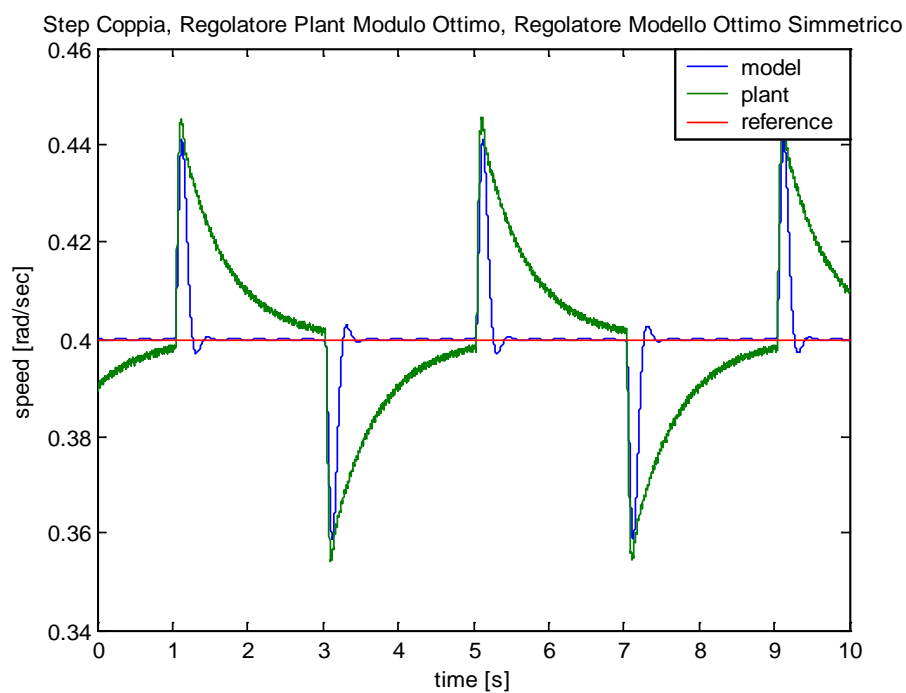
Quindi il guadagno integrale del regolatore di velocità sarà:

$$k_i = \frac{1}{0.092} = 409,8$$

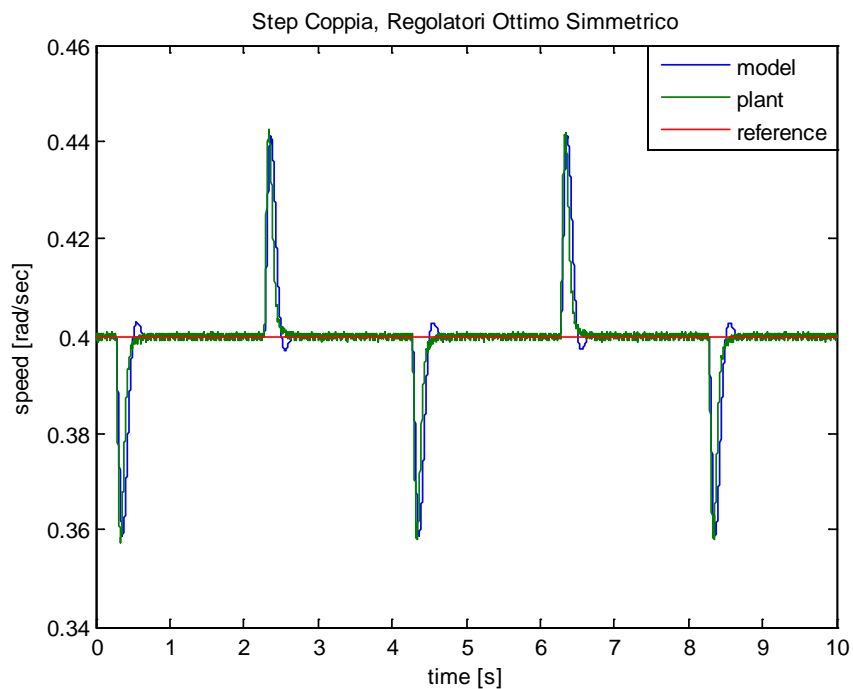
In conclusione, i guadagni del regolatore di velocità, progettato col criterio dell'ottimo simmetrico, sono:

$$k_p = 37,7 \quad e \quad k_i = 409,8$$

Progettato il regolatore col criterio dell'ottimo simmetrico, si testa prima sul modello matematico (figura 5.1.12) e dopo su quello reale (figura 5.1.13). Durante la prova il motore ruota a velocità costante e una coppia di carico ad onda quadra è applicata.



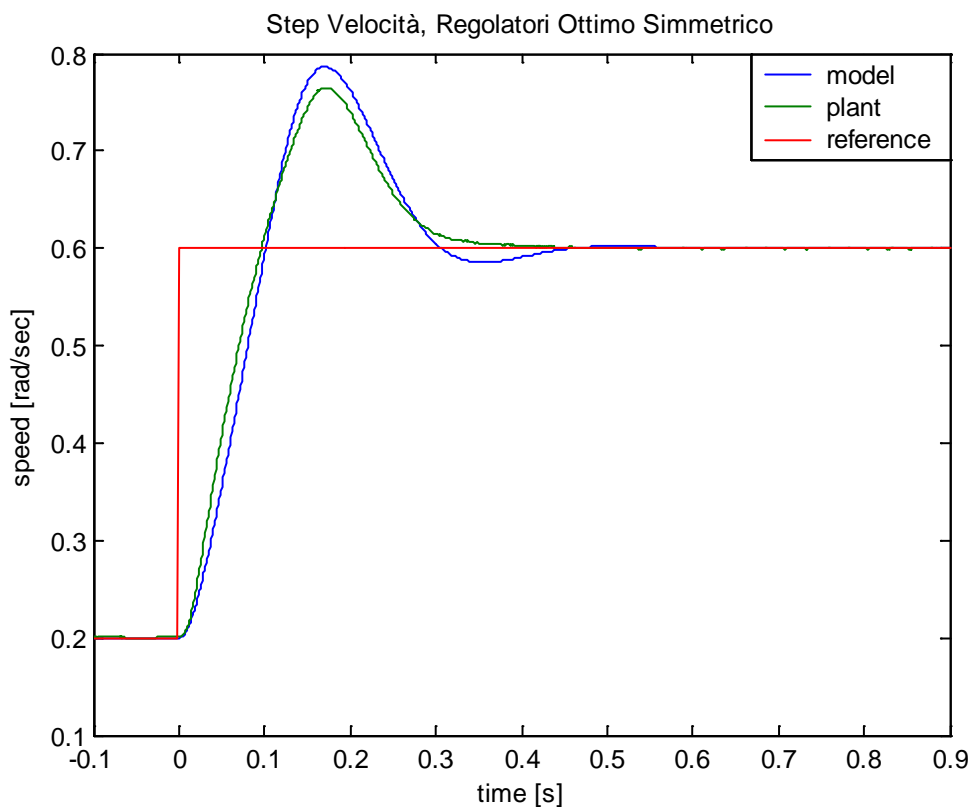
**Figura 5.1.12: Risposta al gradino di coppia con guadagni ottimo simmetrico (nel modello).**



**Figura 5.1.13: Risposta al gradino di coppia con guadagni ottimo simmetrico**

### Filtro di smoothing

Progettato e testato il regolatore di velocità col criterio dell'ottimo simmetrico, si sospende l'applicazione della coppia di carico, e da un set point di velocità costante si passa ad applicare un set point a gradino. In queste condizioni si osserva, che la risposta del sistema al gradino di velocità è caratterizzata da una sovraelongazione molto accentuata (circa il 43 % del valore di regime), che è tipica dei sistemi progettati col criterio dell'ottimo simmetrico (figura 5.1.14).



**Figura 5.1.14: Risposta al gradino di velocità con guadagni ottimo simmetrico.**

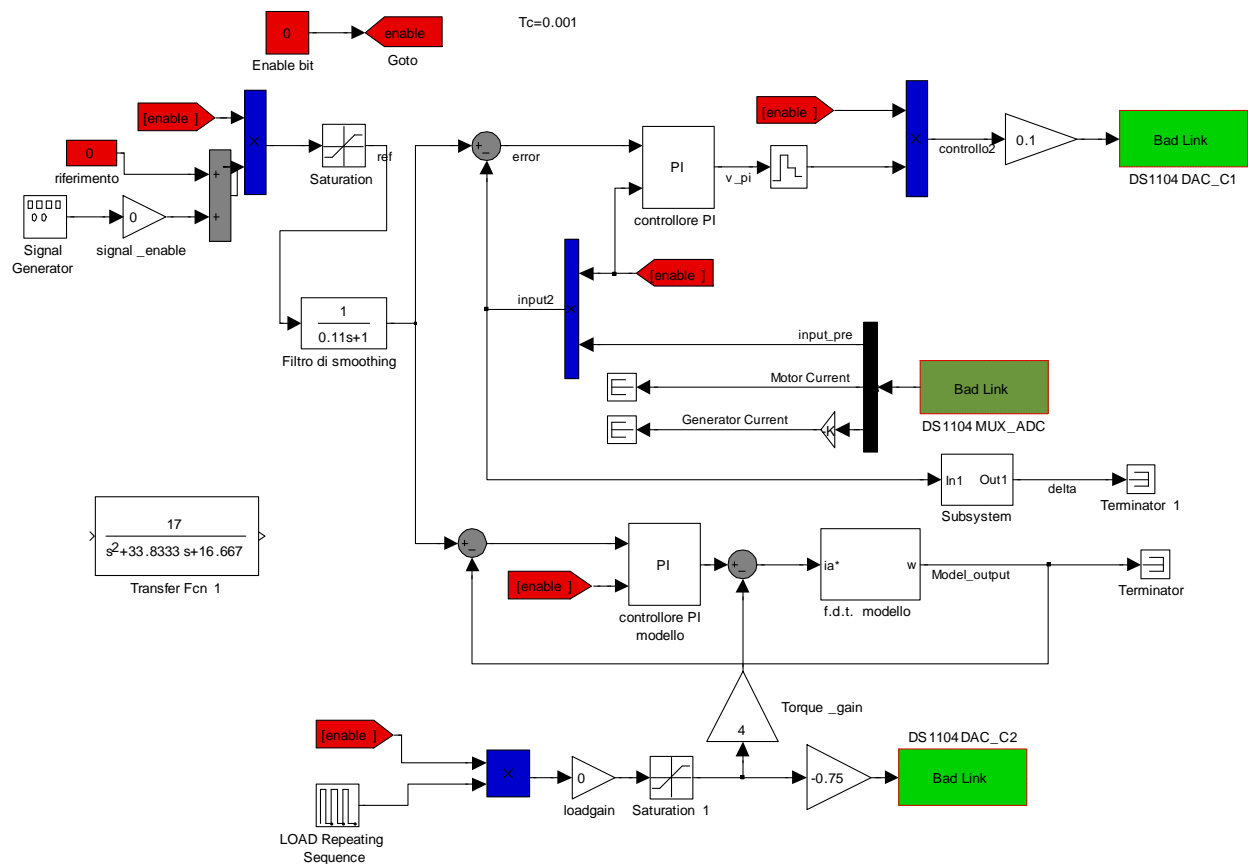
Per ovviare a tale inconveniente, s'inserisce a valle del set point di velocità un filtro di smoothing, la cui funzione di trasferimento è:

$$G_{fs}(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

La costante di tempo del filtro di smoothing ( ) è tarata ad un valore che supera del 20 % la costante di tempo dello zero della funzione di trasferimento del sistema regolatore/plant in anello aperto (l + s). In definitiva si avrà che:

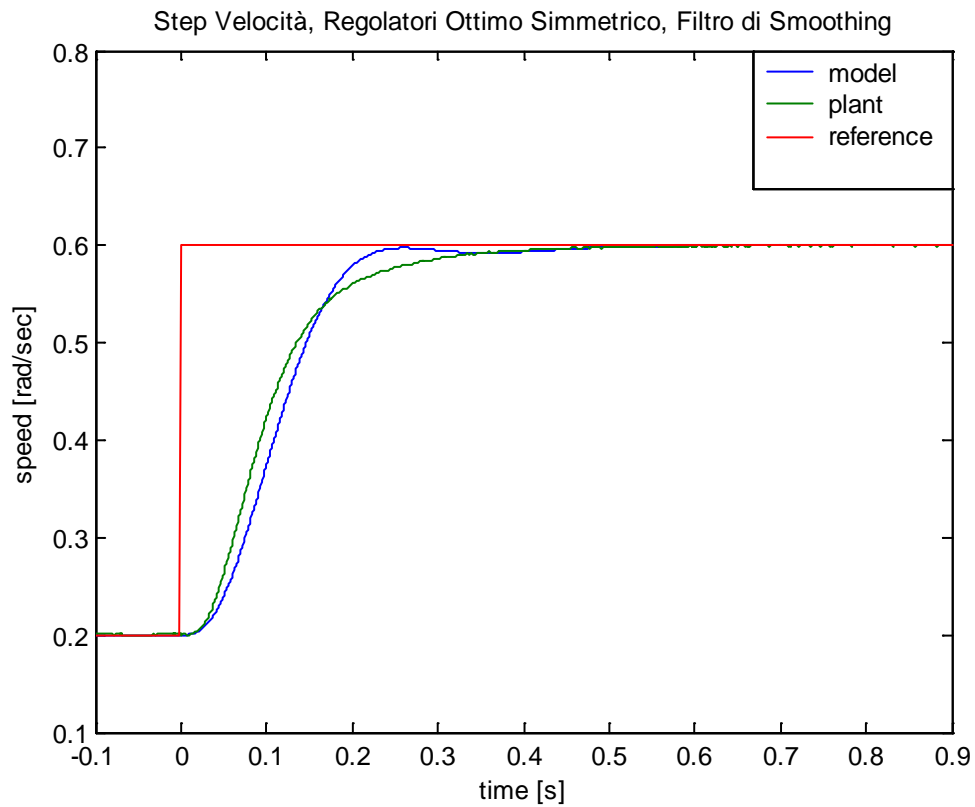
$$= 1,2 \quad = 1,2 \quad = 0,11$$

Introduciamo il filtro di smoothing come si vede in figura 5.1.15:



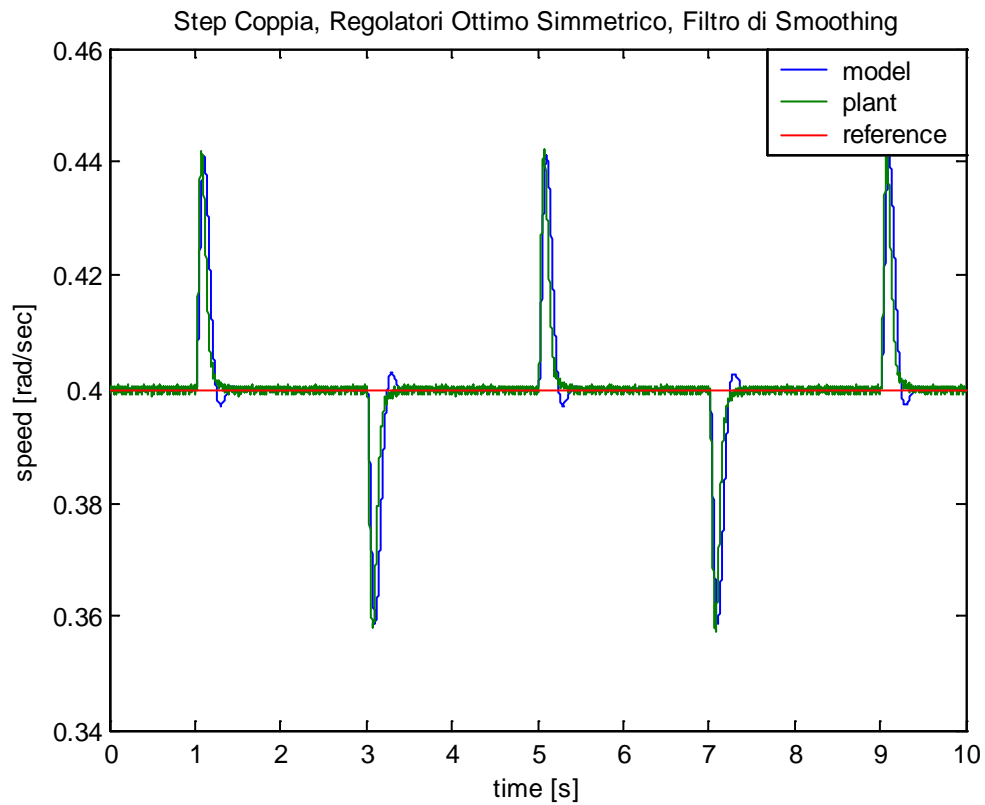
**Figura 5.1.15**

Con la presenza del filtro di smoothing si osserva la risposta del sistema al gradino di velocità (regolatore di velocità progettato col criterio dell'ottimo simmetrico), per verificare l'attenuazione della sovraelongazione (figura 5.1.16).



**Fig 5.1.16: Risposta al gradino di velocità con guadagni ottimo simmetrico e filtro di smoothing.**

A questo punto s'impone un set point costante di velocità (nella prova precedente si trattava di un'onda quadra di velocità) e si applica una coppia di carico (forma d'onda quadra) per verificare la reiezione al disturbo del sistema in presenza del filtro di smoothing. Da questo nuovo test, si osserverà come la presenza del filtro di smoothing non interferisce con la reiezione al disturbo del sistema (figura 5.1.17).



**Fig 5.1.17: Risposta al gradino di coppia con guadagni ottimo simmetrico e filtro di smoothing.**

### Valori di set point

Il set point di velocità è costituito da un offset costante di valore 0,4 a cui può essere sovrapposta un'onda quadra di ampiezza 0,2. L'offset di velocità serve ad evitare che il motore inverta il suo verso di rotazione. Il set point di corrente è un'onda di ampiezza 0,5.



## 5.2 Il controllo di posizione

Con riferimento alle macchine utensili sappiamo, che esse si differenziano in base a due movimenti principali: Di taglio e di avanzamento. Il movimento di taglio ha lo scopo di creare la necessaria velocità relativa tra l'utensile e il pezzo. Il movimento di avanzamento ha lo scopo di alimentare la zona di taglio con nuovo materiale.

Gli azionamenti pilotano gli organi delle macchine utensili, che attuano dei moti di avanzamento, cioè lavorano con frequenti avviamenti e arresti. Per realizzare questi movimenti i singoli azionamenti devono essere controllati in velocità e posizione. I circuiti di regolazione della posizione devono far muovere i singoli elementi secondo dei riferimenti prefissati di posizione e di velocità senza distorsioni o ritardi.

I sistemi di controllo di posizione sono caratterizzati da una configurazione ad anello. Tale configurazione si basa sulla retroazione del segnale di posizione effettivo, rilevato con un dispositivo di misura, che è confrontato con il riferimento di posizione. La differenza tra il riferimento di posizione e il segnale retroazionato è l'ingresso del controllore di posizione.

L'uscita del controllore di posizione è la variabile di comando dell'anello di velocità.

Riprendendo quindi il diagramma a blocchi di figura 5.1.1, aggiungiamo un terzo anello di retroazione per il controllo di posizione, dopo aver integrato la velocità.

Per il controllo di posizione inoltre sarà utilizzato un regolatore di tipo proporzionale  $K_p$ , che avrà il compito di produrre un'uscita quanto più sovrasmorzata possibile.

Per le nostre simulazioni è stato utilizzato un blocco *Simulink* per la generazione di una traiettoria definendo quindi sia un set point di velocità, di accelerazione che di posizione.

Possiamo vedere il diagramma già modificato in figura 5.2.1:



La interfaccia grafica creata per lo schema in Simulink, possiamo vederla in figura 5.2.2:

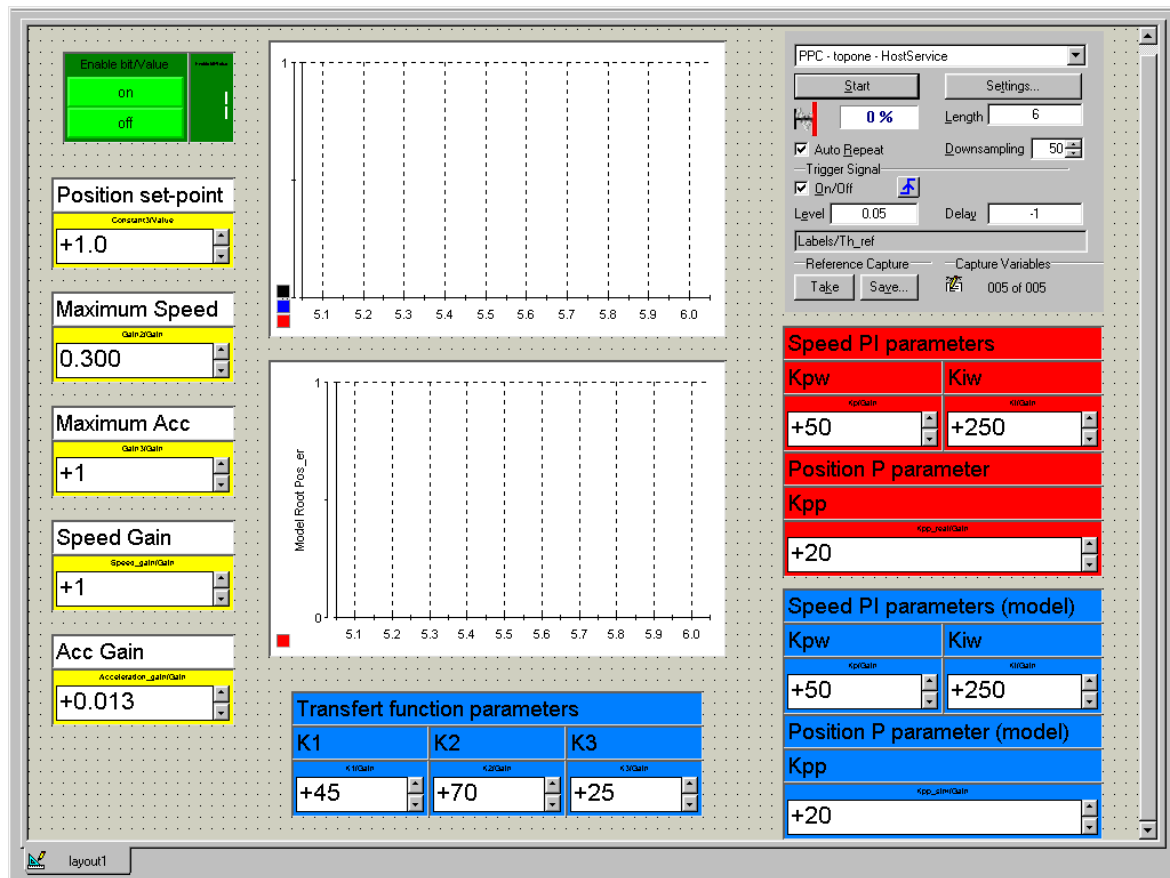
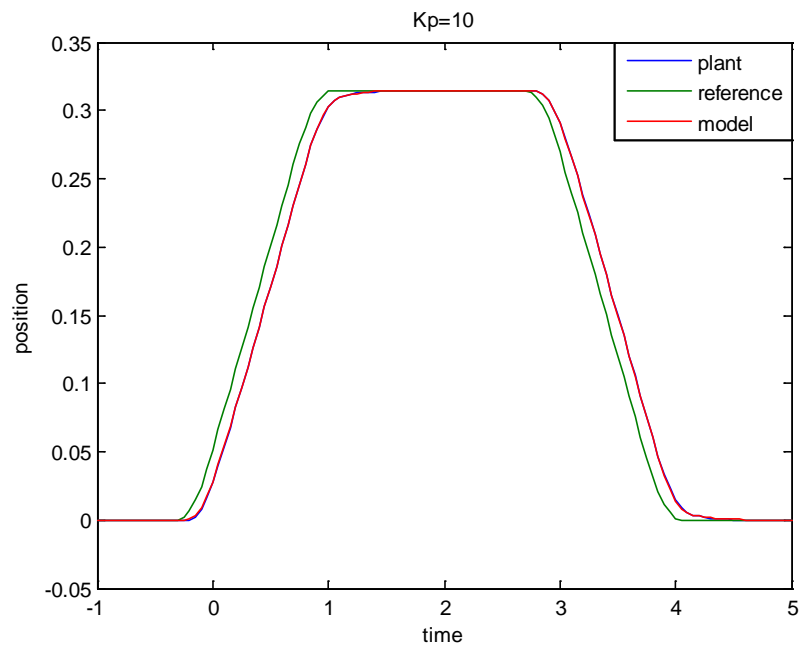
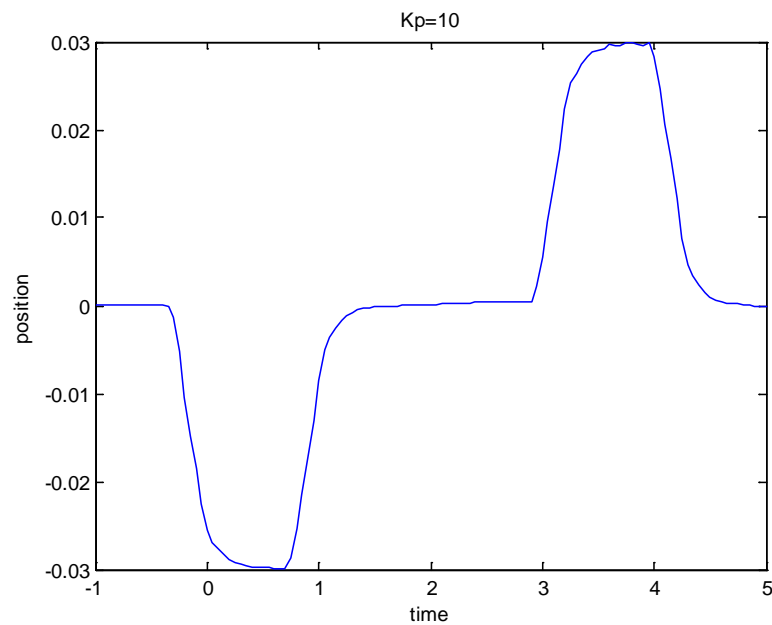


Figura 5.2.2

Effettuiamo una prima prova per  $K_p=10$  con un *set point* di posizione di  $10 \cdot 2\pi \text{ rad}$  e i risultati ottenuti sono:

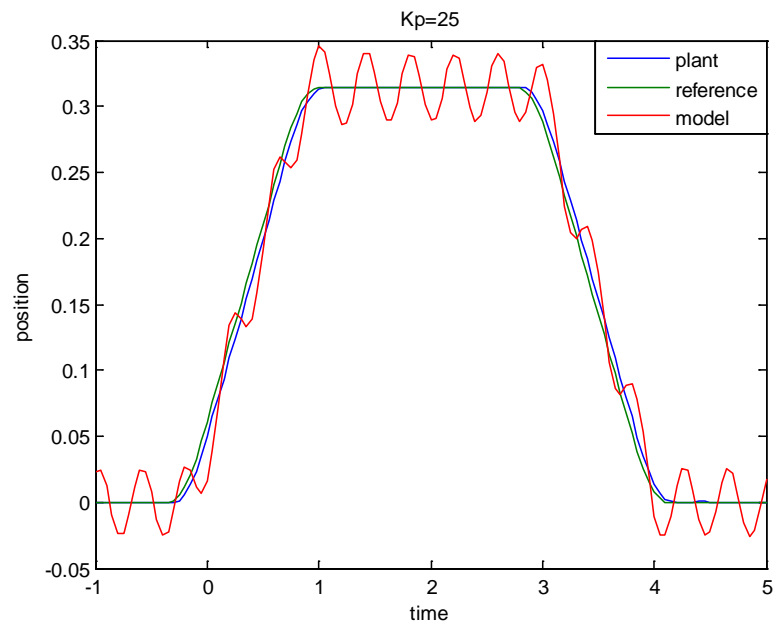


**Figura 5.2.3**

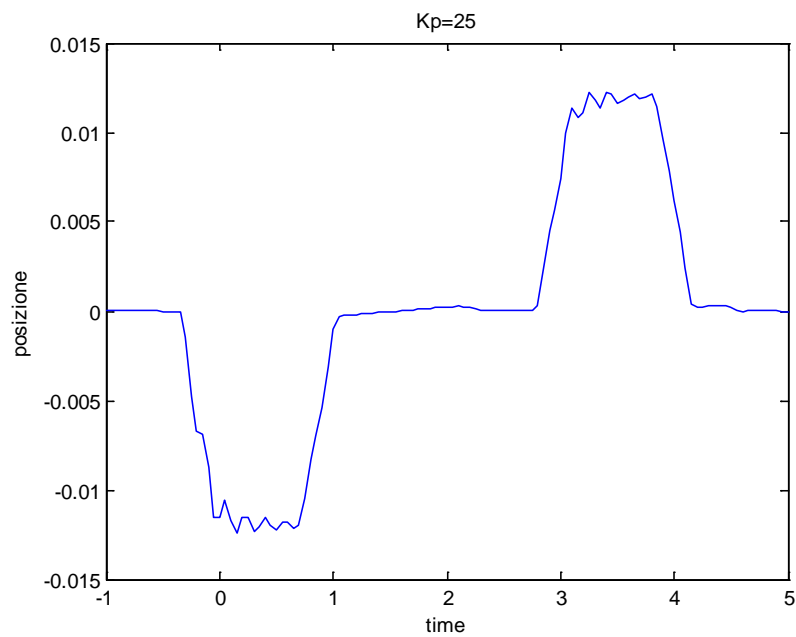


**Figura 5.2.4**

Si vede che la stabilità è buona, ma l'errore può essere ridotto. Per una  $K_p=25$ :



**Figura 5.2.5**

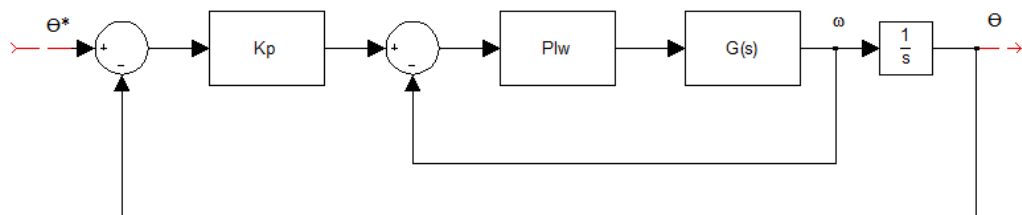


**Figura 5.2.6**

L'errore è diminuito notevolmente, ma il sistema è adesso più vicino a condizioni di instabilità.

Per trovare una  $K_p$  adeguata utilizzeremo il Luogo delle Radici.

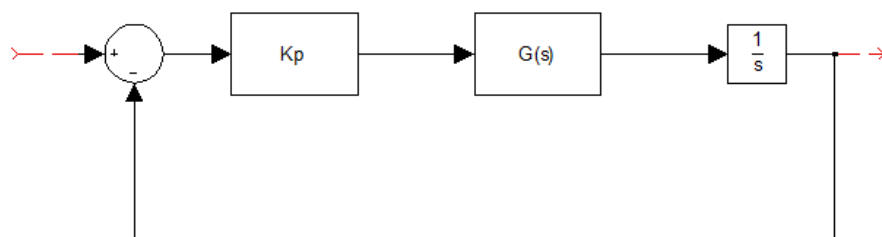
In primo luogo dobbiamo trovare la funzione di trasferimento del sistema, considerando  $K_p=1$ :



**Figura 5.2.7**

$$G_T(s) = P I_\omega * G(s) = (K_{p\omega} + \frac{1}{s}) * \frac{1}{s} = (38 + 410/s) \frac{1}{s} = \frac{38s + 410}{s^2}$$

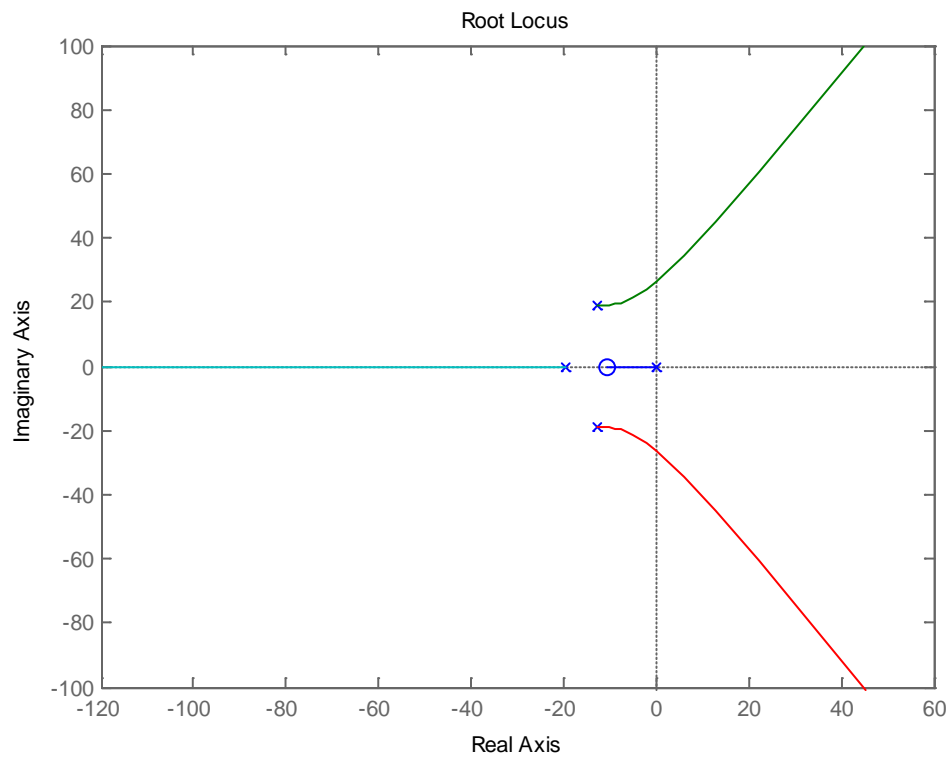
$$G_0(s) = \frac{1}{s}$$



**Figura 5.2.8**

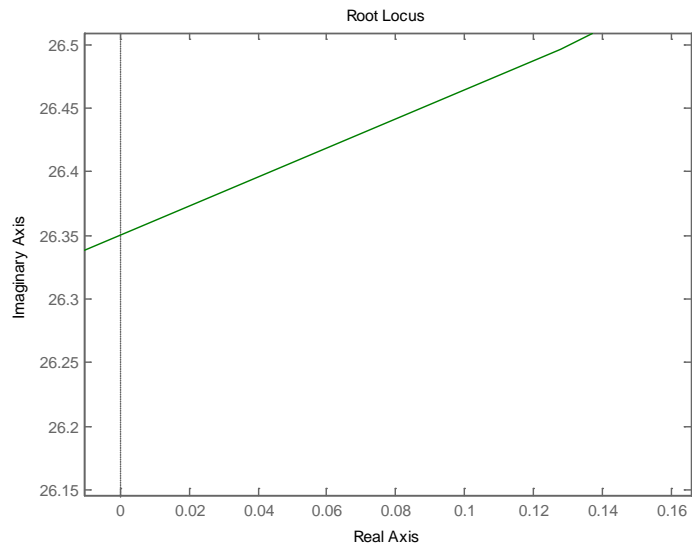
$$G_F(s) = \frac{1}{s}$$

Da questa funzione tracciamo il luogo delle radici:



**Figura 5.2.9**

Facendo *zoom* in figura 5.2.9 possiamo vedere (Figura 5.2.10) il valore del guadagno limite per la stabilità del sistema;



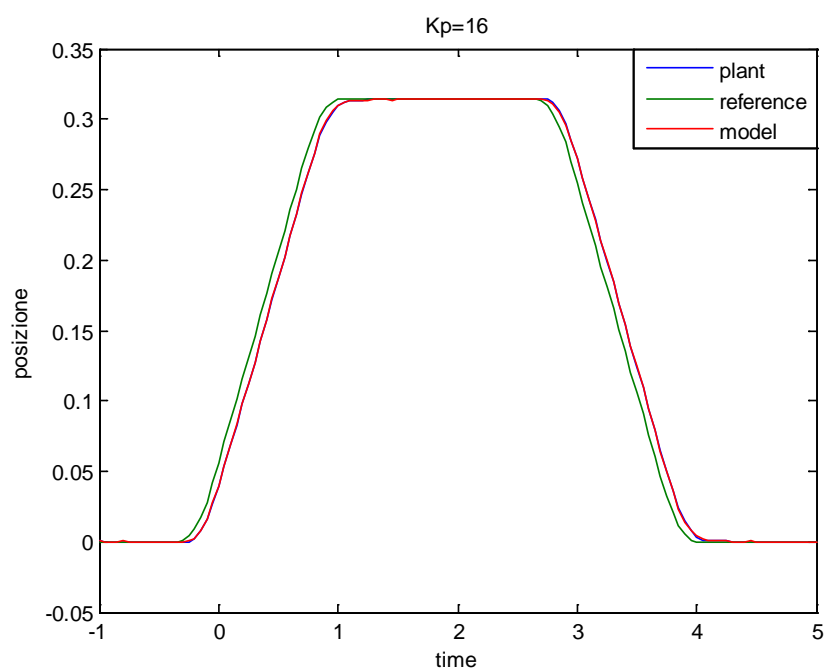
**Figura 5.2.10**

Questo valore è  $K_0=26.35$ . Per assicurare la stabilità del sistema  $K_p$  sarà il 60% di  $K_0$ . Quindi:

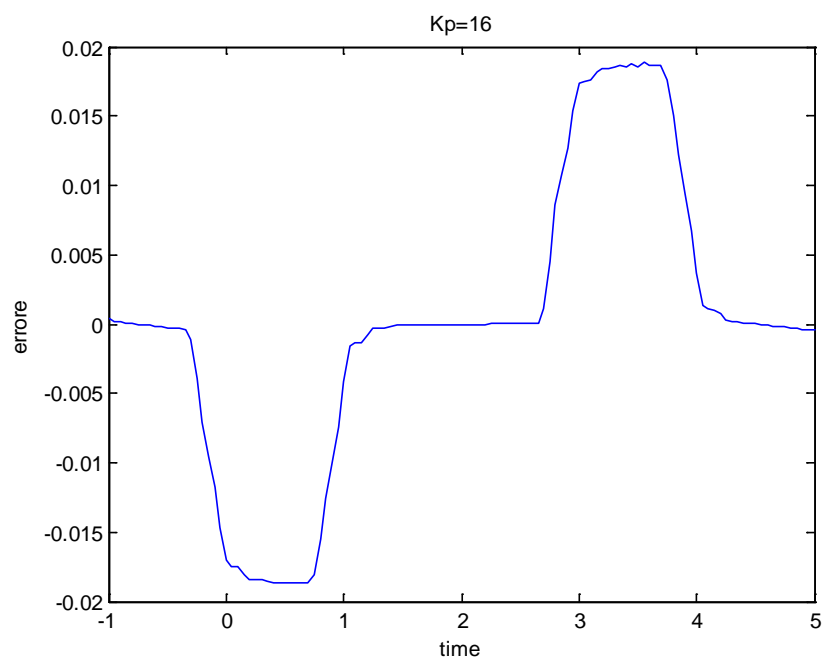
$$K_p = 0,6 * K_0 = 0,6 * 26,35 = 15,81 \approx 16$$

Questo sarà il valore utilizzato per il nostro regolatore proporzionale. Possiamo vedere che così abbiamo buona stabilità e un errore accettabile:





**Figura 5.2.11**



**Figura 5.2.12**

Il lavoro di questa tesi è stato quello di dimostrare come sia possibile effettuare un controllo di velocità di un motore in corrente continua utilizzando un controllo in cascata con due anelli di retroazione, e con due regolatori PI per il controllo degli anelli di velocità e di corrente, e anche come fare il controllo di posizione aggiungendo un terzo anello di retroazione.

Le prove sono state svolte presso il laboratorio di robotica del DEE con l'ausilio di un laboratorio sperimentale costituito dal motore, dalla varia sensoristica (*Amira DR300*), e dai vari dispositivi elettronici per l'interfacciamento in ambiente *Matlab* per una simulazione in *real time* (Scheda *DS1104*).

È, stato inoltre realizzato un diagramma a blocchi dettagliato riferendoci alla scheda tecnica del motore, il quale potrebbe descrivere, con un margine di errore apprezzabile, i vari dispositivi che compongono l'azionamento.

Un successivo lavoro da effettuare su questa tesi, potrebbe essere quello di ottimizzare maggiormente la funzione di trasferimento ottenuta, tenendo conto dei ritardi intrinseci dei dispositivi di misura, di fenomeni di non linearità dovuti ai dispositivi elettronici che compongono l'azionamento, e di un coefficiente di attrito che in realtà è non lineare. Quindi effettuare un controllo non lineare del sistema.

## Appendice A

### The dSPACE DS1104 R&D Controller Board

Designing a feedback control system typically requires several steps:

- Identifying a dynamic model from experimental data.
- Use of computer aided-design tools to construct a compensator.
- Verification of the design using a computer simulations.
- Implementation of the compensator.

While MATLAB and the SIMULINK block diagram environment are useful for control design and analysis, the dSPACE DS1104 R&D Controller Board provides the means for acquiring system identification data and implementing a discrete-time controllers for analog plants. In Laboratory #2, the dSPACE system is used to generate the actuating signal to the PA150C Pre-Amplifier input and to record the resulting tachogenerator and Output Potentiometer responses in a system identification experiment. In Laboratory #5 you will use the dSPACE system to implement a discrete-time controller for a continuous-time plant.

The dSPACE system consists of three components: the DS1104 controller board mounted within a personal computer, a breakout panel for connecting signal lines to the DS1104 controller board, and software tools for operating the DS1104 board through the SIMULINK block diagram environment. Figure 17 shows a block diagram of the DS1104 controller board.

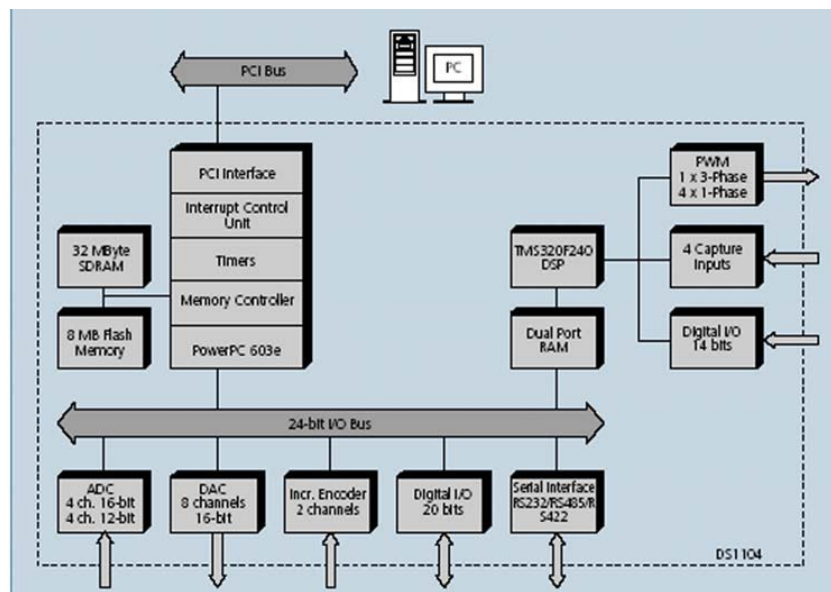


Figure 17: Block diagram of the DS1104 controller board mounted within the personal computer.

Of primary interest in Laboratory #2 are the digital-to-analog (D/A) converters and the analog-to-digital (A/D) converters. The DS1104 has eight, 16-bit, D/A converters each with an output range of  $\pm 10$  V. The D/A converters operate as voltage sources as their output impedance is close to zero, and each D/A channel can source or sink up to 5 mA. The resolution, which represents the smallest difference between two output levels of the converter, is  $20 \text{ V}/2^{16} = 0.305 \text{ mV}$ . That is, the output voltage of the D/A converter can be varied from -10 V to +10 V in 0.305 mV steps. The DS1104 also contains eight A/D channels. The first four channels share a single 16-bit A/D converter through an analog multiplexer. As a result, it is not possible to simultaneously sample analog signal on these four channels. The remaining four channels each have a dedicated 12-bit A/D converters, allowing the simultaneous sampling of four analog input signals. The input

range for all eight channels is  $\pm 10$  V. The resolution of the 16-bit channels is 0.305 mV, while the resolution for the 12-bit A/D converters is  $20 \text{ V}/2^{16} = 4.9 \text{ mV}$ . The input impedance of each A/D channel is approximately  $1 \text{ M}\Omega$ .

Figure 18 shows the breakout panel for the dSPACE DS1104 R&D Controller Board. Connections to the A/D and D/A converters are made using BNC connectors. The D/A channels are located at the lefthand side of the breakout panel, and are labeled DACH1 through DACH8. Immediately to the right of the D/A output jacks are eight BNC connectors labeled ADCH1 through ADCH8 for the A/D converters. Connections between the breakout panel and the MS150 Servo system are made using BNC to banana-jack coaxial cables. It is important to ground the black banana jack to the MS150 Servo system for each BNC cable connected to the breakout panel.



Figure 18: Signal connections to the dSPACE DS1104 R&D Controller Board mounted within the personal computer are made through the breakout panel shown above.

Operation of the dSPACE DS1104 board is controlled through SIMULINK. In Laboratory #2 you will need to download two files from the EE 428 web page. The first file, sysidexp.mdl, is the SIMULINK block diagram shown in Figure 19. The second file, DataCapture.m, is a m-file that executes the SIMULINK block diagram in Figure 19, displays the data in a single MATLAB figure, and saves the data to a MATLAB file for further analysis. Figure 20 shows the contents of the m-file.

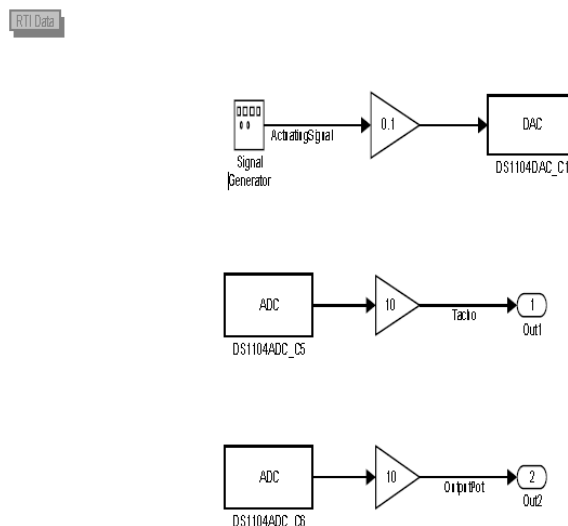


Figure 19: SIMULINK block diagram for the system identification experiments.

```

% EE 428 Fall 2005
% Lab 2
% m-file for controlling system identification experiment
% close open figures and clear the workspace
close all
clear
% select processor board 'ds1104'
mlib('SelectBoard','ds1104');
% select variables to be traced and obtain their descriptors
var_names = {'simState','Model Root/ActuatingSignal';...
'Model Root/Tacho';'Model Root/OutputPot';'currentTime'};
var = mlib('GetTraceVar',var_names);
% stop the processor by setting simState to 0
mlib('Write',var(1),'Data',0);
% set the data acquisition options
mlib('Set','Trigger','ON',...
'TriggerLevel',0.1,... % Triggers when current Time = 0.1 sec
'TriggerEdge','rising',... % default, can be omitted
'TriggerVariable',var(5),... % trigger on variable 'currentTime'
'TraceVars',var,...
'NumSamples',99000,... % 99000 samples
'Delay',0);
% start the processor by setting simState to 2
mlib('Write',var(1),'Data',2);
% start capture on DS1104
mlib('StartCapture');
% wait until capture is done
while mlib('CaptureState') ~= 0, end
% fetch data after capture is complete
out_data = mlib('FetchData');
Ea = out_data(2,:);
Et = out_data(3,:);
Ep = out_data(4,:);
t = out_data(5,:);
% plot results
subplot(3,1,1);
plot(t, Ea);
title('PA150C Pre-Amplifier Input Voltage');
ylabel('Voltage [V]');
subplot(3,1,2);
plot(t, Et);
title('Tachometer Output');
ylabel('Voltage [V]');
subplot(3,1,3);
plot(t, Ep);
title('Output Potentiometer');
ylabel('Voltage [V]');
xlabel('Time[s]');
% save data to file
save sys_id_data t Ea Et Ep

```

Figure 20: MATLAB m-file for controlling the system identification experiments.

In order to use the dSPACE board, first download sysidexp.mdl and DataCapture.m to a working directory on your P drive. Next, start MATLAB version 7 and change the **Current Directory** displayed at the top of the MATLAB command window to the directory that contains the files sysidexp.mdl and DataCapture.m. Type **SIMULINK** on the command prompt to open the SIMULINK block library. Select **Open** from the SIMULINK library window's **File** menu and then choose the model file sysidexp. SIMULINK will open and display the block diagram of sysidexp.mdl as shown in Figure 19.

The block diagram in sysidexp.mdl represents the process that is downloaded to, and executed by, the dSPACE DS1104 R&D Controller Board. The **RTI Data** box in the upper lefthand corner indicates that the model is associated with the real-time interface of the dSPACE system. The **SIMULINK Signal Generator** block produces a discrete-time square-wave signal that drives the PA150C Pre-Amplifier in the MS150 Servo system via a zero-order hold D/A converter (DS1104DAC c1, digital-to-analog converter channel 1). This D/A block, as well as the two A/D blocks that sample the output voltage of the tachogenerator and output potentiometer unit, are selected from the real-time interface block set directory shown in Figure 21. SIMULINK automatically scales the input of the D/A converter so that a  $\pm 1$  input range maps to a  $\pm 10$  V D/A output range. Similarly, the A/D inputs are scaled so that a  $\pm 10$  V input range is mapped into a  $\pm 1$  range within SIMULINK. To compensate for this scaling, three SIMULINK **gain** blocks appropriately weight the signal input to the D/A converter and the signal outputs from the A/D converters.

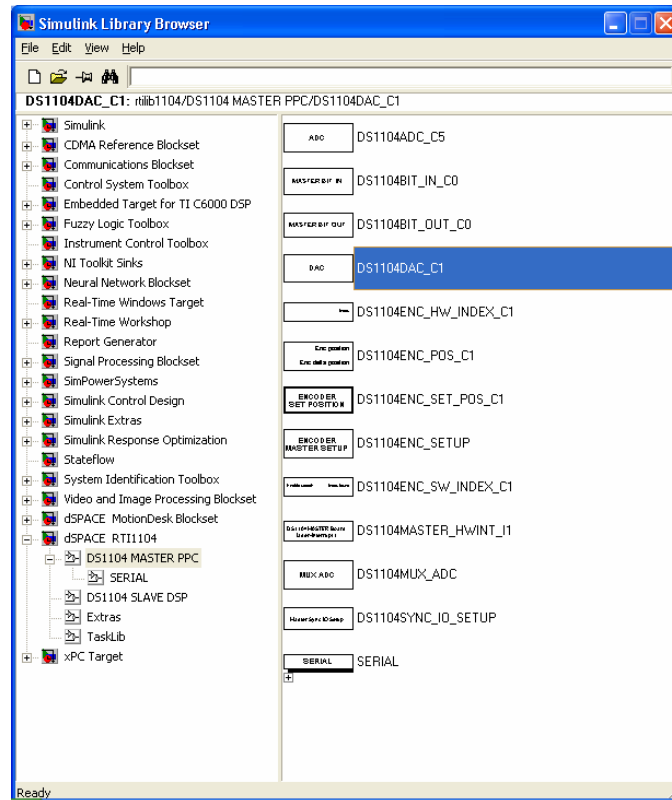


Figure 21: Block sets served by the processor on the dSPACE DS1104 controller board.

To view and adjust the sample rate and experiment duration, you must first open the solver configuration window. Select the **Configurations...** option from the **Simulation** menu as show in Figure 22. Next, select **Solver** from the tree-structured view of the **Configuration Parameters**. Figure 23 shows that the experiment has duration time of ten seconds and uses a fixed sample rate sample rate of (1/10 kHz) seconds. You cannot change these values unless you also modify the m-file DataCapture.m, as this file assumes these values.

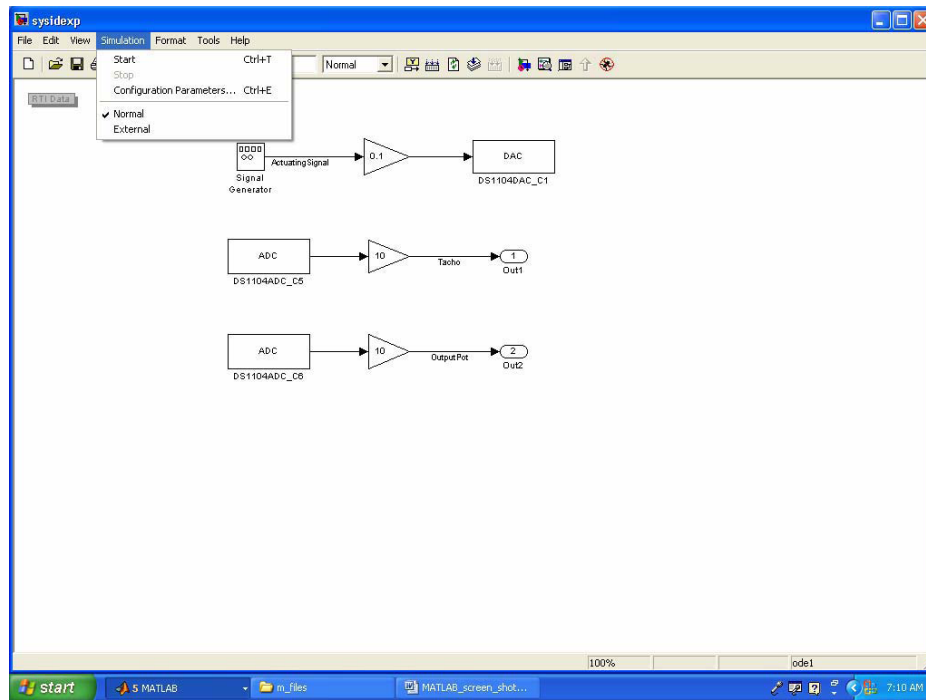


Figure 22: Select the Configuration Parameters menu.

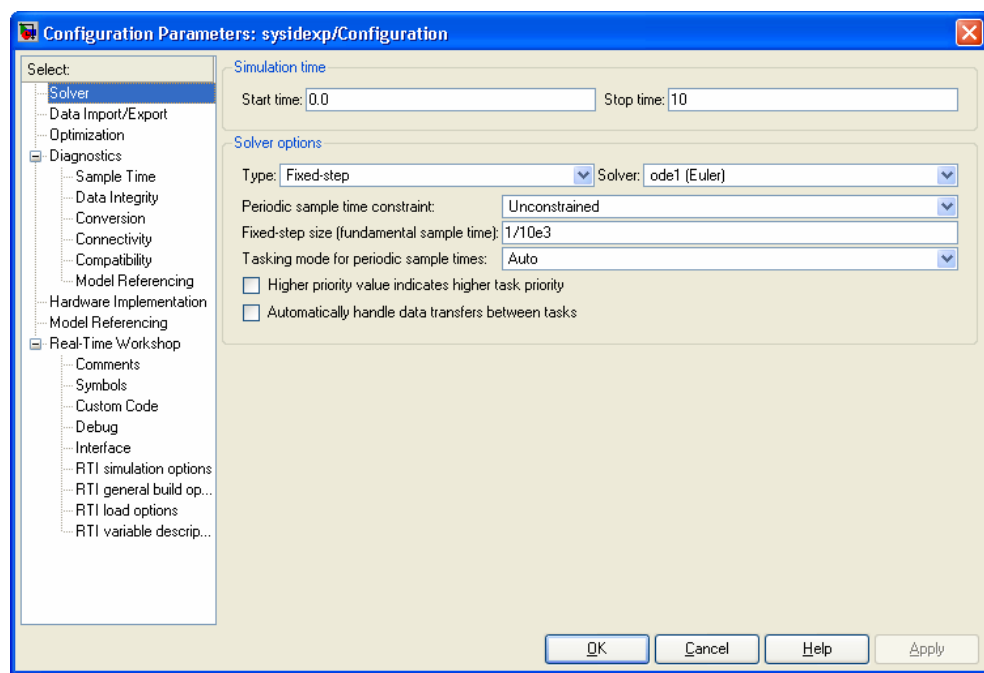


Figure 23: The experiment duration and sample rate is set within the **Solver** dialog box.

The dSPACE D/A dialog box, that is opened by double clicking on an D/A converter, allows you to specify the channel number, the initial value of the output, and the final value of the output as shown in Figure 24. It is important to set the D/A output to zero volts on termination of the experiment, otherwise the output shaft of the MS150 DC motor may continue to rotate after the system identification experiment is complete. The dSPACE A/D blocks have a single dialog window that allows specification of the channel number.

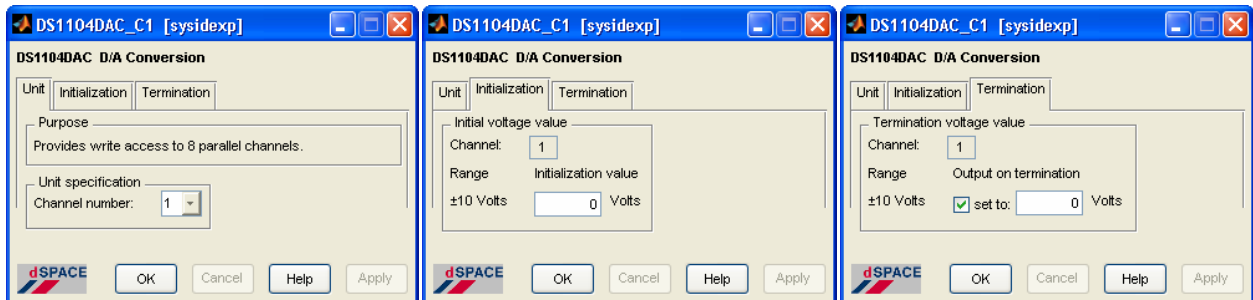


Figure 24: Dialog windows for selecting D/A parameters.

Figure 25 shows the block parameters for the signal generator block. A square waveform, with a peak amplitude of 0.01 V and frequency of 0.5 Hz, is selected. The frequency is chosen so that the DC motor speed reaches a steady-state value during the interval when the square-wave output is constant, thereby facilitating estimation of the DC gain of the process. The amplitude is chosen so that the PA150C Pre Amplifier does not saturate.

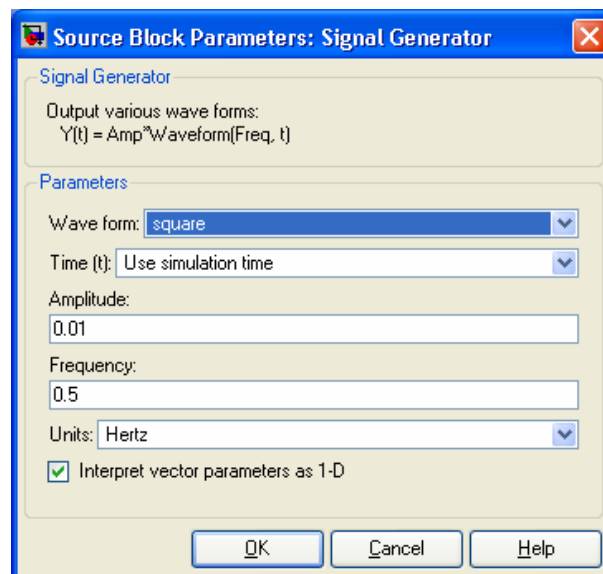


Figure 25: Dialog box for specifying the waveform, amplitude, and frequency of the signal generator block.

The signals ActuatingSignal, Tacho, and OutputPot in the SIMULINK block diagram represent the driving voltage to the PA150C Pre-Amplifier unit, the output voltage of the tachogenerator, and the output voltage of the Output Potentiometer unit, respectively. These variables are accessed by the m-file DataCapture.m.



In order to implement the experiment represented by the SIMULINK block diagram, you must click on the incremental build icon shown in Figure 26. After selecting this command, observe the sequence of operations performed by the dSPACE software in the MATLAB command window. The dSPACE software first translates the SIMULINK block Diagram into C code, and then compiles the C code to form an executable file for the processor residing on the dSPACE DS1104 board. Finally, the dSPACE software downloads the executable code to the processor on the DS1104 board. If you change any parameters within the SIMULINK block diagram, for example, the amplitude of the square wave, you must perform another incremental build to update the model parameters on the dSPACE controller board. In Laboratory #5, you will learn how to change the parameters on the controller board from a windows application.

There are two mechanisms for controlling execution of code on the DS1104 processor. The dSPACE **Control Desk** application provides a graphical user interface for controlling, modifying, and automating experiments implemented with the DS1104 controller board. Figure 27 shows an example of a Control Desk window. The flexibility of the Control Desk environment comes at the expense of a timely learning curve for using Control Desk features. For this reason, Laboratory #2 uses a less sophisticated, but simpler approach to execute the system identification experiment and upload data from the dSPACE board into the MATLAB environment. In Laboratory #5 you will learn how to work with the Control Desk environment.

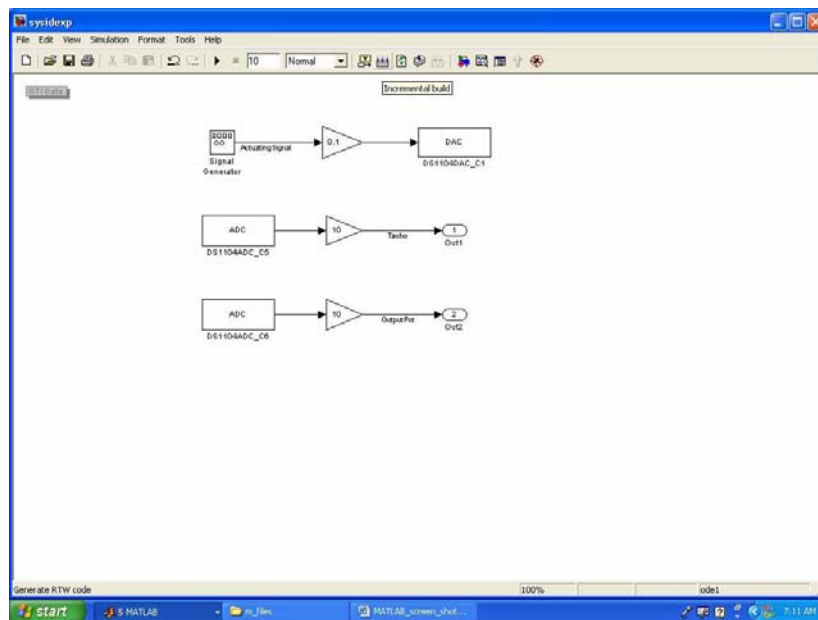


Figure 26: Click on the incremental build button to download the SIMULINK block diagram to the processor on the dSPACE DS1104 controller board.

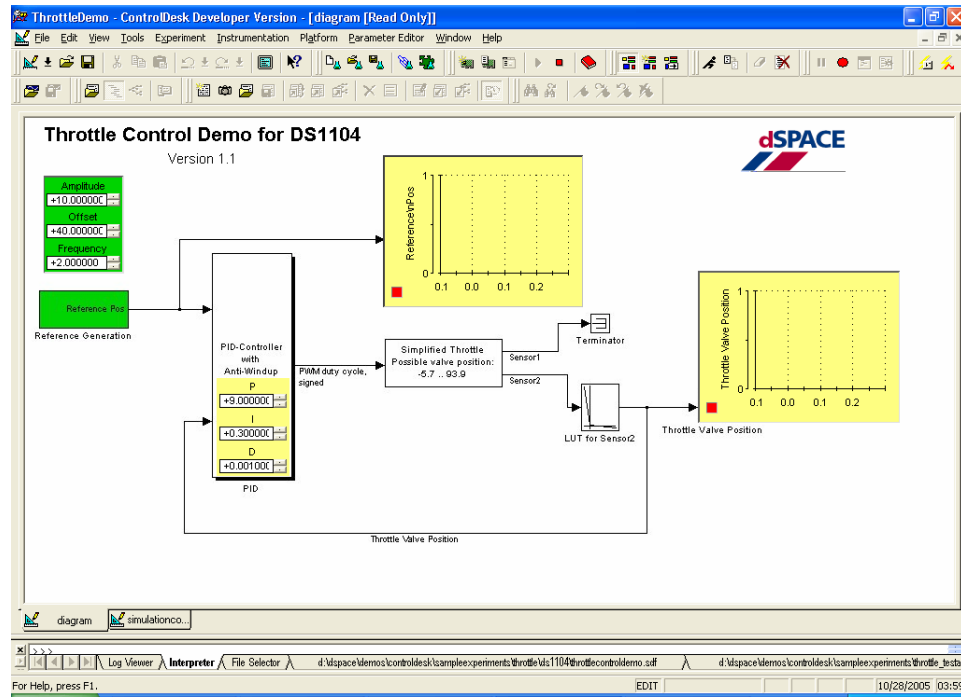


Figure 27: The Control Desk application provides a graphical user interface for adjusting dSPACE DS1104 controller parameters during experiments. In this example, the proportional-plus-integral-plus-derivative gains for an engine throttle control system may be varied while the experiment is in progress.

The dSPACE function **mlib** provides a command line function for controlling the operation of the dSPACE DS1104 processor directly from MATLAB, and is the key function used in the m-file DataCapture.m shown in Figure 20. This m-file has three tasks. First, it commands the dSPACE DS1104 processor to start the system identification experiment. Second, it polls the dSPACE DS1104 processor to determine when the system identification experiment is complete. Third, at the end of the system identification experiment, it uploads the data from the dSPACE processor board, displays the data in a single MATLAB figure, and then saves the data to the file sys\_id\_data.mat. The sampling instants, the driving voltage to the PA150C Pre-Amplifier unit, the output voltage of the tachogenerator, and the output voltage of the Output Potentiometer unit are saved in the vectors  $t$ ,  $E_a$ ,  $E_t$ , and  $E_p$ , respectively.

## Appendice B

### CONTROL DESIGN FOR SERVO AMIRA DR300

Jirka Roubal ✉ Petr Augusta ✉  
Vladimir Havlena ✉ Jindrich Fuka ✉

*Department of Control Engineering, Faculty of Electrical  
Engineering, Czech Technical University in Prague  
Karlovo namesti 13, 121 35 Prague, Czech Republic  
E-mail: (roubalj, augusp1, havlena, fuka) @control.felk.cvut.cz*

**Abstract:** This paper is written for students who are beginning to study at Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague as a motivation for studying control engineering. Therefore, the below mentioned references are introduced with regard to this fact. The paper shortly presents procedures of the control design, including a description of a system, an identification of its parameters, a simple and an advanced controller design. The controller design procedures are illustrated in the control design for the laboratory model servomechanism AMIRA DR300 that is held in laboratory K26, Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague.  
**Keywords:** Systems, System identification, PID Controllers, Kalman filters, LQG control.

#### 1. INTRODUCTION

There are many different processes in practice and we usually want to control their properties, for instance, we want the rotation speed of a motor to be equal to some value for any load torque, we want planes not to fall down, we want to increase the electric power of power plants with decreasing the air pollution, we want to increase the capacity of hard discs so we have to control the reading machinery more accurately etc.

The regulation is based on feedback control, see Figure 1. The output of a system  $y(t)$  is measured by a sensor. The controller computes the input of the system  $u(t)$  based on the measured output  $y(t)$  and its reference  $w(t)$  and applies this value by an actuator to the system such that the output behaves how we want i. e. how we set the reference.

For example, the brain observes our position by eyes and stimulates our muscles so we do not fall down. For a control design, we have to know how our system behaves; we do the identification Ljung, L. (1987); Horacek, P. (1999) and describe a system by mathematical tools usually. Very often, we do a physical setup and get a system of differential (recurrent) and algebraic equations at first. The second step is to determine the system inputs and outputs, to derive a so called *model* from mathematical equations of the system and, if needed, to linearize it Horacek, P. (1999). Then

we can measure the system parameters and write down the complete model with numerical values. There are two main approaches to linear systems in the control theory: state-space methods Kailath, T. (1980) and a polynomial description Kucera, V. (1979). The former uses four matrices and is often written as a system of differential (recurrent) and algebraic equations.

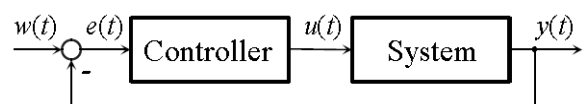


Figure 1. The feedback control loop

The latter expresses a linear model as a transfer function. *Single Input Single Output* (SISO) system is described by a fraction of two polynomials, i. e. ratio of Laplace transform of the output and Laplace transform of the input with zero initial conditions in continuous-time case and ratio of Z-transform of the output and Z-transform of the input with zero initial conditions in discrete-time case. *Multi Input Multi Output* system is described by a fraction of two polynomial matrices.

When we have the model of our system we can design a controller. There are many ways how to design the controller. The controller design depends on what we want to achieve, for

example, stability of the system and quality of the system behavior, optimal behavior of the system according to a criterion, etc.

This paper presents identification of servomechanism AMIRA { DR300 (see Figure 2) which is held in laboratory K26, Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University (CTU) in Prague.

Then several controllers are designed and applied to the system, for example, PID controller Franklin, G. F. et al. (2002); John, J. (1996), LQ controller Astrom, K. J. and Wittenmark, B. (1997); Havlena, V. (2001) with Kalman filter Kailath, T. et al. (2000); Havlena, V. (2002); Krokavec, D. and Filasova, A. (2002). The outline of the paper is as follows: in Section 2, the servomechanism AMIRA DR300 is described. In Section 3, model identification is performed. Section 4 presents the application of several controllers for servomechanism AMIRA DR300.

## 2. SERVO AMIRA DR 300

Servo AMIRA DR 300 (see Figure 2) is a servomechanism consisting of two identical motors which are connected by a mechanical clutch.

The first motor is used for control of the rotation speed or the shaft angle. The second one, in the following called as a generator, is used for a simulation of load torque.

The whole system consists of three components. The first part is an I/O card MF614 (Humusoft (<http://www.humusoft.cz/>)) with analogue and digital inputs and outputs. The second, power part, contains sources, current sensors and amplifiers. The third part contains two motors and sensors of the rotation speed and the shaft angle.

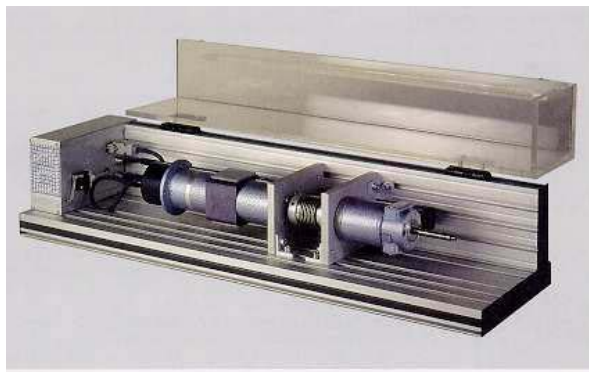


Figure 2. The servomechanism AMIRA DR 300

The system is controlled by the Real Time Toolbox of MATLAB (<http://www.mathworks.com/i>). Note that the motor is relatively small and its behavior is nonlinear for low values of input voltage; its mathematical description by the linear control theory is very inaccurate. Therefore, we control the rotation speed in the linear area only.

## 3. IDENTIFICATION OF THE SYSTEM

As it has been said, we have to know how our system behaves for a control design; we have to have a model of the system. This part of the control theory is taught at the CTU in Prague in the subject Systems and Models Horacek, P. (1999); Fuka, J.; John, J.; Kutil, M. (<http://dce.felk.cvut.cz/sari/>).

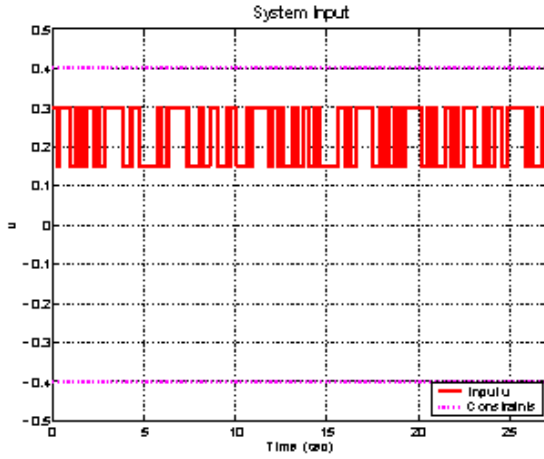
For control, we consider one input (voltage of the motor) and one output (rotation speed). In this section, we introduce differential equations which can describe behavior of a direct current (DC) motor. Then we identify a transfer function from the input voltage to the output rotation speed of the motor.

### 3.1. Mathematical Description of a DC Motor

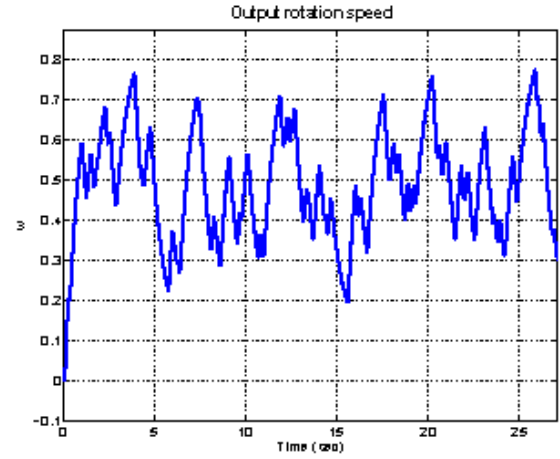
A direct current motor with a permanent magnet can be described by differential equations

$$\begin{aligned} \frac{di(t)}{dt} &= -\frac{R}{L} i(t) - \frac{k}{L} \omega(t) + \frac{1}{L} u(t), \\ \frac{d\omega(t)}{dt} &= \frac{k}{J} i(t) - \frac{1}{J} M_z(t), \end{aligned} \quad (1)$$

where  $i(t)$  [A],  $\omega(t)$  [rad/s],  $u(t)$  [V],  $M_z$  [N\*m] are the current of the motor, rotation speed, input



(a) The input signal for the system identification



(b) The output response to the input identification signal

Figure 3. The input and output signals for the identification of the transfer function of the system

voltage and load torque respectively, and  $R$ ,  $L$ ,  $k$ ,  $J$  are motor parameters. Equations (1) are called, in the control theory, state space model Horacek, P. (1999).

As it has been said above, we consider that the voltage  $u(t)$  is the input (manipulated variable) of the system and the rotation speed  $\omega(t)$  is the output (controlled variable) of the system. Instead of continuous-time state space description (1), we can use a transfer function in Laplace variable  $s$  (complex variable) Horacek, P. (1999). This transfer function is expressed as a fraction of Laplace transform of the output rotation speed and Laplace transform of the input voltage

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{\frac{k}{LJ}}{s^2 + \frac{R}{L}s + \frac{k^2}{LJ}}. \quad (2)$$

### 3.2. Identification of the System AMIRA DR 300

For the identification of the system, we perform some experiments. At first, we generate a suitable input signal, see Figure 3(a), apply it to the system and measure the output rotation speed, see Figure 3(b). Note that the input signal provides operation in a linear area and excites the system inside a sufficient frequency band. Using the least squares (LS) method Luenberger, D. G. (1996); Horacek, P. (1999); we can compute

the transfer function coefficients from the input and output sequences in Figure 3.

The resulting continuous transfer function is

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{247.5}{(s + 33.98)(s + 1.402)}. \quad (3)$$

Now we must validate if model (3) is correct and sufficiently accurate. By comparing the step responses of the system and its model (3), see Figure 4, we can see that our model is very accurate.

Note that the step response of the system is measured in the linear area; the motor is spun and when the rotation speed reaches some steady-state value, the step change of the input voltage was applied to its input.

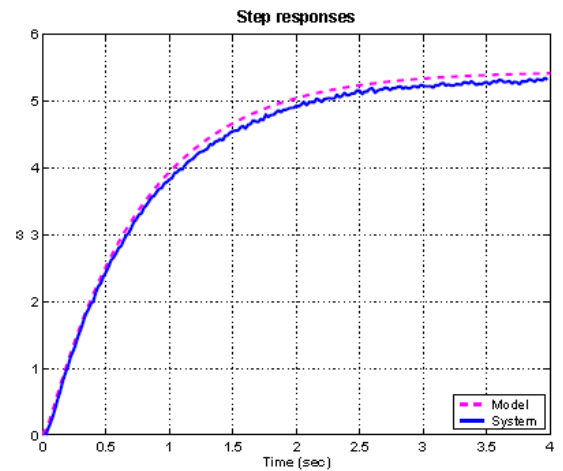


Figure 4. The step responses - system and its model

There are many identification methods. The method that we choose depends mainly on the characteristics of a system. For example, it is not possible to identify a nuclear reactor by using the input sequence from Figure 3(a).

#### 4. CONTROLLER DESIGN FOR THE SYSTEM

As it has been said, we can design many controllers for a system. For the design, we can use either a state space description or a transfer function. We can design either a continuous controller consisting of integrators, amplifiers and summators or a discrete controller which is usually represented by a digital computer. In this section, we shortly describe the main principles of several controllers and apply these controllers to the servomechanism AMIRA DR300.

The comparison of these regulations is shown. As it has been said above, we consider one manipulated variable (input voltage  $u$ ) and one controlled variable (output rotation speed  $\omega$ ).

##### 4.1. PID Controller

The PID block is a continuous controller consisting of three blocks: proportional, integral and derivative John, J. (1996). The design of the PID controller is based on a continuous transfer function of the system. For the design, we can use for example frequency methods, the root locus method John, J. (1996), etc.

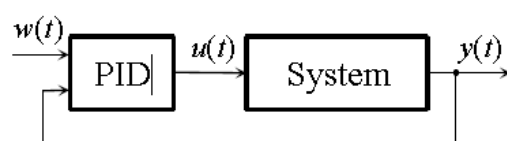


Figure 5. The closed loop with the PID controller

The closed loop behavior with the PID controller are shown in Figure 6. Note that in approximately 6 seconds, the step change of the generator input was applied to the servomechanism AMIRA DR300 which simulates the change of the load torque.

The theory of PID controllers is taught at the CTU in Prague in the subject Systems and Control John, J. (1996); Fuka, J.; John, J.;

Kutil, M. (<http://dce.felk.cvut.cz/sari/>).

##### 4.2. LQ Controller with Kalman Filter

The main idea of an LQ controller design is the minimization of a quadratic criterion which is weighting the square of a manipulated variable and the square of a controlled variable Havlena, V. (2001). LQ means a linear system and a quadratic criterion.

In this paper, we use the discrete LQ controller design based on a state space description. The inputs of this controller are system states, but we measure only the output rotation speed. So we use the Kalman filter Havlena, V. (2001) for estimating these states. The Kalman filter is the optimal state observer Kailath, T. (1980); Stecha, J.; Havlena, V. (1999) whose design is based on covariance of noise Havlena, V. (2002). The inputs of the Kalman filter are system inputs and system outputs, and the outputs of the Kalman filter are estimations of system states.

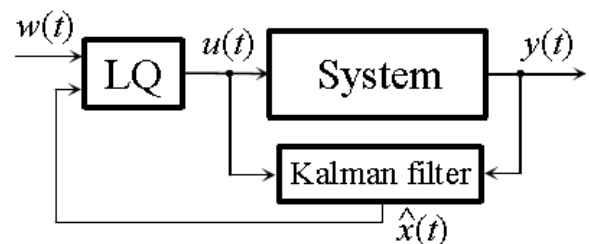


Figure 7. The closed loop with the LQ controller

The closed loop behavior with the LQ controller and Kalman filter are shown in Figure 8. If we observe Figure 8(b) carefully, we discover that, unlike in PID control (Figure 6(b)), the controlled variable (output rotation speed) does not asymptotically track the reference signal, because the steady state errors of the estimates of the system states are not equal to zero. This problem is solved in the next subsection.

The theory of dynamical systems and analysis of the dynamical systems are taught at the CTU in Prague in the subject Theory of Dynamical Systems Hurak, Z.; Hromcik, M.; Roubal, J. (<http://dce.felk.cvut.cz/tds/>); Stecha, J.; Havlena, V. (1999).

The theory of the LQ controller design and the Kalman filter design are taught at the CTU in Prague in the subject Modern Control Theory Havlena, V. (2001); Roubal, J.; Pekar, J. (<http://dce.felk.cvut.cz/mtr/>).



#### 4.3. LQ Controller with Kalman Filter + with the Load Torque Estimation

As it has been said above, the classical Kalman filter is not able to reach zero errors of the estimates of the system states. The reason of this is that the Kalman filter inputs are only the system input and system output. The Kalman filter does not know about a load torque which is caused by friction or by the generator (the second motor of the servomechanism AMIRA DR300). Therefore, the Kalman filter design is modified such that the Kalman filter not only estimates system states, but the load torque too, see Figure 10.

The closed loop behavior with the LQ controller and modified Kalman filter are shown in Figure 9. You can see in Figure 9(b) that in this case, the controlled variable asymptotically tracks the reference signal.

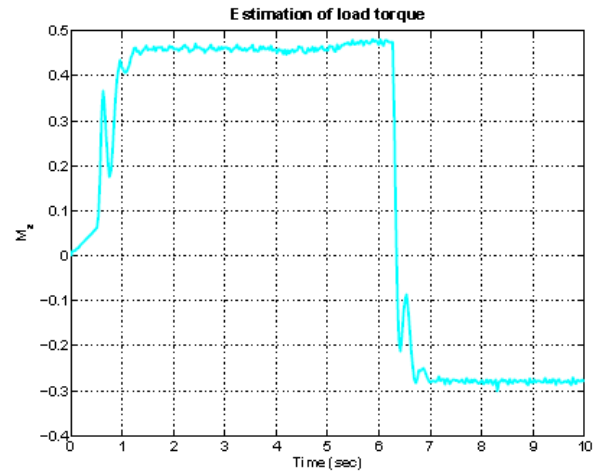


Figure 10. The estimation of the load torque

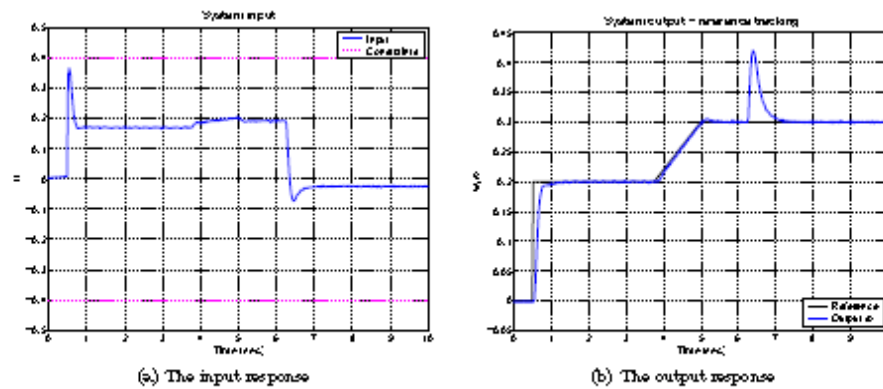


Figure 6. The input and output responses - closed loop behaviour with the PID controller

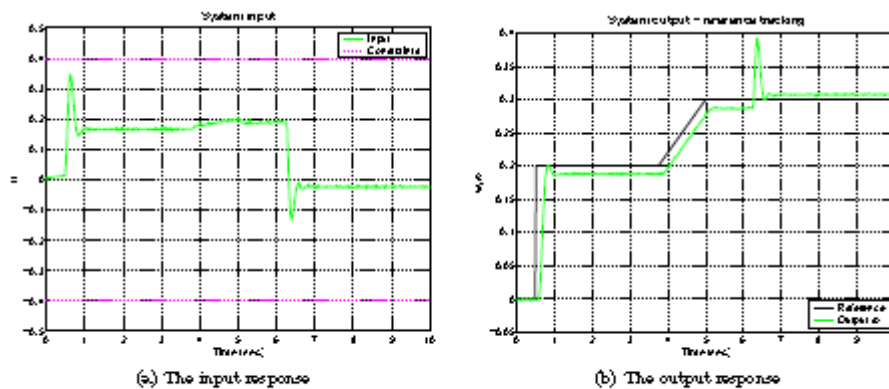


Figure 8. The input and output responses - closed loop behaviour with the LQ controller and the Kalman filter

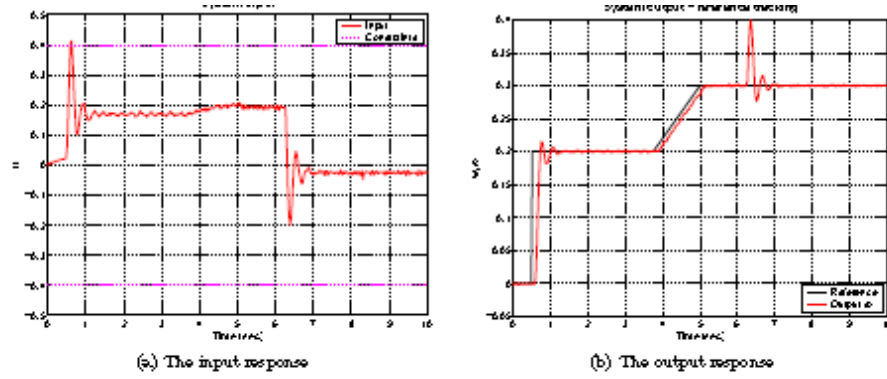


Figure 9. The input and output responses – closed loop behaviour with the LQ controller and Kalman filter + the estimation of the load torque

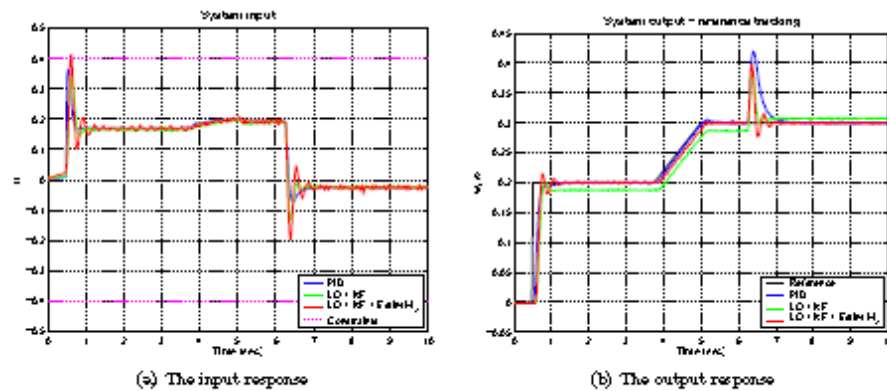


Figure 11. The input and output responses – closed loop behaviour – comparison of the controllers

## 5. CONCLUSION

This paper has been written for students as a motivation for studying control engineering. The control design for laboratory model AMIRA DR300, including a description of the system, identification of its parameters, a simple controller design and an advanced controller design, is presented.

For the system identification, the least squares method is used and the transfer function between input voltage and the output rotation speed is

obtained. The model is used for the PID controller design and the LQ controller design with the Kalman filter.

The classical Kalman filter is not able to reach zero steady state of the output estimation error, the system output does not track our reference signal. Therefore the Kalman filter is modified for estimating of the load torque of the servomechanism. The comparison of particular closed loops are shown in Figure 11.



## **Bibliografia**

- [1] Dispense di “Azionamenti elettrici” di Prof. Ing. Francesco Cupertino.
- [2] Dispense di “Accionaminetos Eléctricos” di Jesús López Taberna.
- [3] José Basilio Galván Herrera, “Control Analogico de Sistemas Linealies”, Ulzama Digital.
- [4] dSpace. Digital signal processing and control engineering GmbH. DS1104 User’s Guide- Floating Point Controller Board.
- [5] Luis Guasch Pesquer, “Máquinas y Accionamientos Eléctricos”, Marcombo.